ESD-TR-88-125

Technical Report
808

Neural Network Performance on the Stochastic Exclusive-or Problem

R.Y. Levine

8 July 1988

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Department of the Army
under Electronic Systems Division Contract F19628-85-C-0002.

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This program is sponsored by the U.S. Army Strategic Defense Command under Air Force Contract F19628-85-C-0002.

The views and conclusions contained in this document are those of the contractor and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the United States Government.

The ESD Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Hugh L. Southall

Hugh L. Southall, Lt. Col., USAF
Chief, ESD Lincoln Laboratory Project Office

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

NEURAL NETWORK PERFORMANCE ON THE STOCHASTIC EXCLUSIVE-OR PROBLEM

R.Y. LEVINE
Group 93

TECHNICAL REPORT 808

8 JULY 1988

LEXINGTON MASSACHUSETTS

ABSTRACT

The application of neural networks to the detection of variance transitions in Gaussian noise is analyzed. The problem is a benchmark example of hypothesis testing on a nonstationary stochastic process. Comparisons among perceptron, back propagation nets, and classical (Neyman-Pearson) decisioning are provided by Monte Carlo simulation. False alarm and detection probabilities for classical and neural net algorithms are computed and compared.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

NEURAL NETWORK PERFORMANCE ON THE STOCHASTIC EXCLUSIVE-OR PROBLEM

I. INTRODUCTION

In this report the application of neural networks to the detection of variance transitions in Gaussian noise is considered. The problem, which consists of transition detection between a pair of input sample variances, is a benchmark example of hypothesis testing on a nonstationary stochastic process. In the case of neural net algorithms the testing of hypotheses results from the decision space output of the last layer of neurons.

Variance transition detection in a Gaussian random process is probably the most tractable example upon which to study machine analysis of a nonstationary stochastic process. It is an obvious test bed for the analysis of neural network decisioning of stochastic data. In this report the modeling of the process as Gaussian is dictated by the desire for a comparison of neural network and classical detection techniques. For specific applications, other data-derived parameters, such as correlation length, may be more appropriate for the analysis of transitions. It is expected, however, that neural network structures required for hypothesis testing are independent of the nature of the sufficient statistics; dependent instead upon the pattern of mean values. The purpose of this research is to provide a theoretical foundation for work currently being done on satellite maneuver detection. The detection of maneuvers is performed by neural networks which have been trained upon signal variances, spectral widths, and autoregressive model coefficients from radar cross section data. This research is also relevant to the data fusion effort, in which these parameters are obtained from different sensors.

The definition of a neural net is summarized in Figure 1. It consists of a series of parallel input lines connected to a set of layered neurons.[1] Each neuron performs a prescribed (generally nonlinear) operation on the set of values applied simultaneously to the input, and outputs a value to the right along a connection. The only function of the connections is multiplicative weighting. Overall net output from the last layer of neurons consists of elements in a binary decision space reflecting various hypotheses about the data. Knowledge about the mapping between data and decision spaces is stored in the connection weights $\{W_{ij}\}$ shown in Figure 1. These values are adjusted by a learning algorithm which uses data for which correct hypotheses have been identified.[2] It is exactly the distributed storage of information (connection weights) which suggests that powers of association and robustness of human intelligence may be realized in neural net algorithms.

The fundamental problems of neural net design involve net structure and learning algorithms. A particular mapping can be realized by infinitely many net structures. Although work has been done to identify properties of net functioning which depend upon structure, particularly in the area of learning time[2] and memory capacity,[3,4] the overall principles of structure which

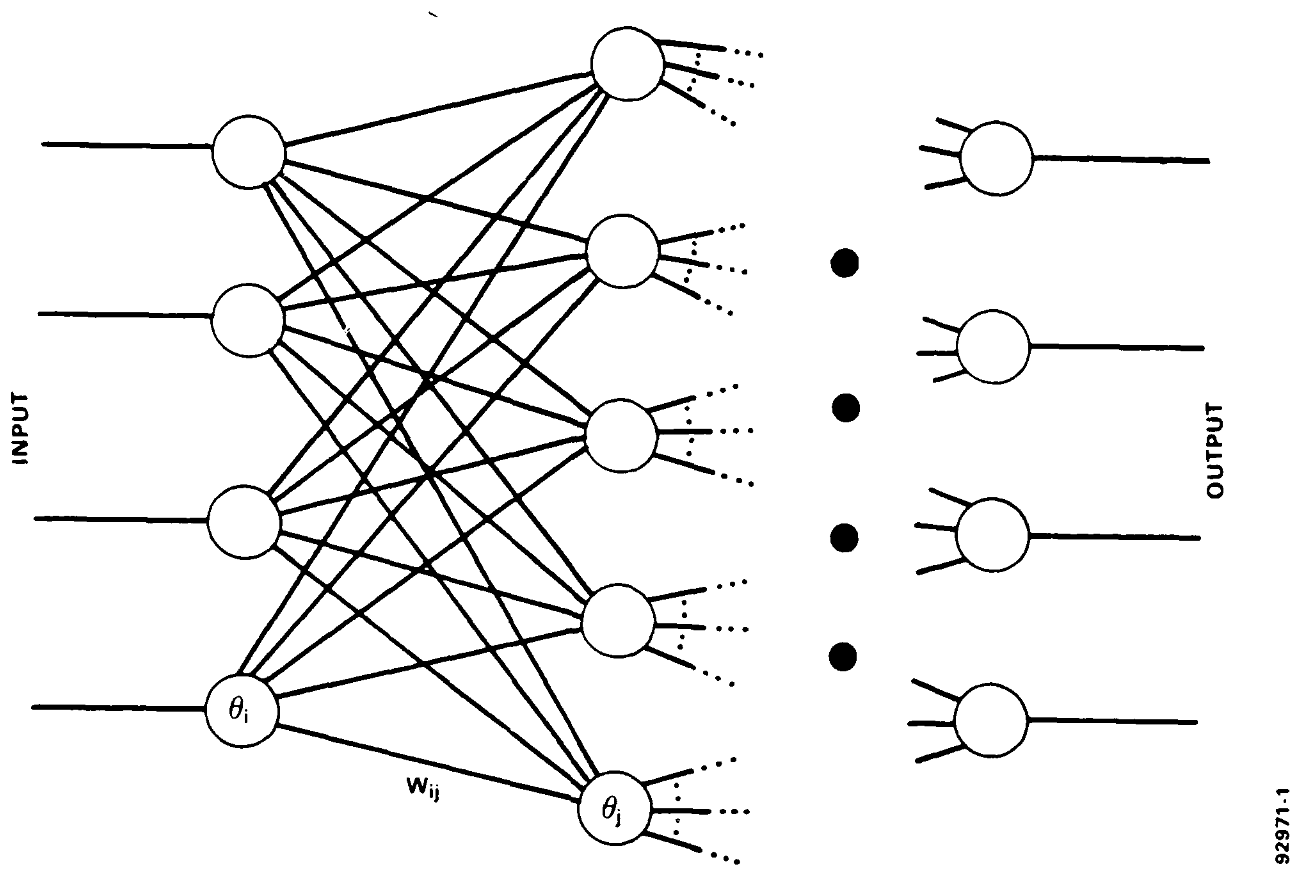


Figure 1. Generic neural net: Connection weights $\{W_{ij}\}$ for neurons i and j; thresholds $\{\theta_i\}$ for each neuron i.

govern function have yet to be discovered. An issue which arises in hypothesis testing, and precedes the problem of neural net structure, is the comparison of neural net and classical test performance. While it is probable that neural net analysis succeeds in areas of decisioning in which classical techniques fail, the quantitative comparison of tests for problems amenable to both techniques deepens an understanding of the possibilities of neural hypothesis testing.

In this report a Gaussian random process is considered in which a transition of the variance is possible. Sample variances before and after the supposed transition form a pair of χ^2 distributed sufficient statistics which are input to the net. The input of sufficient statistics in fact embodies the application of prior information about the process to aid net decisioning. In principle, entire data sequences could be input to the net, which would then be forced to discover features of the data (such as sufficient statistics) upon which to base decisions. Net operation upon sufficient statistics is a simple relevant example of modular neural net design; that is, prior rule structure (compute sufficient statistics) grafted upon parallel neural decisioning (test hypotheses).[5]

Section II of this report consists of a review of classical hypothesis testing for the case of transition detection in a Gaussian random process. False alarm and detection probabilities are expressed in terms of the threshold parameter used in the hypothesis test. The optimum threshold, corresponding to maximum detection and minimum false alarm probabilities, is computed for a number of different noise and sampling conditions. These probabilities form the basis for the comparison of neural network and classical decisioning.

False alarm and detection probabilities for neural networks are ensemble-derived from Monte Carlo simulation. Starting with a random set of connection weights $\{W_{ij}^{(0)}\}$, the networks were trained on a stochastic data set by application of known learning algorithms (either perceptron convergence or back propagation). False alarm and detection probabilities were computed directly from the proportion of successful mappings obtained on a performance ensemble of stochastic data. Sections III and IV contain the results of the Monte Carlo analysis of perceptron[6,7] and back propagation networks,[2] respectively. Transition detection for various noise conditions and sampling windows is studied as a function of the applied number of iterations of the learning algorithms. Section V contains quantitative comparisons between classical (Neyman-Pearson) and neural network performance. This is obtained from the asymptotic (large iteration number) values of P_f and P_d for neural networks, and the corresponding classical probabilities at the optimum threshold.

II. VARIANCE TRANSITIONS IN GAUSSIAN NOISE

As mentioned in the introduction, an issue which precedes net design for hypothesis testing is the quantitative comparison of neural net and classical decisioning. This is possible in a Gaussian process in which an identified sufficient statistic such as variance exists. In this section the Neyman-Pearson test for variance transitions is derived, and relevant conditional probabilities are numerically computed.[8]

The classical approach to transition detection for a Gaussian process requires the definition of the statistic,[8]

$$X = \sum_{i=1}^{N} y_i^2 \quad , \tag{1}$$

where $\{y_i | i = 1,2,\ldots,N\}$ is the zero mean stochastic data in a window of length (N). The variable X is χ^2-distributed with a probability density

$$p(x) = \frac{x^{(N/2-1)} \exp\left[-x/2\sigma^2\right]}{2^{N/2}\sigma^N \Gamma(N/2)} \tag{2}$$

where x is the standard deviation of the Gaussian random variable y. In the case of two possible deviations, σ_0 and σ_1, classical hypothesis testing results from the definition of a threshold γ which determines the decision; that is, x greater (less) than γ implies noise deviation σ_1 (σ_0).

The computation of false alarm and detection probabilities for transition detection requires the definition of conditional probabilities $\{p[(i,j) | (l,m)] \,|\, i,j,l,m \in (0,1)\}$. Each pair (i,j) corresponds to a (before, after) variance condition, where an index i of 1 or 0 denotes a high or low variance, respectively. The conditional probability p[(i,j) | (l,m)] represents the detection of noise condition (i,j) when the windows truly correspond to deviation pairs (l,m). Assuming independent tests of each data window, the conditional probability factorizes

$$p[(i,j) \,|\, (l,m)] = p(i|l)\; p(j|m) \quad , \tag{3}$$

where p(i|j) denotes the probability of choosing deviation i with a true noise deviation j. The pair of decisions required to analyze a transition is based upon the values of x in Equation (1) for two data windows and the threshold γ (as described above).

Detection and false alarm probabilities for transition detection can be expressed in terms of single segment hypothesis test probabilities by application of Equation (3). It is shown in the Appendix that the transition hypothesis test probabilities are given by

$$P_d = p(\text{transition}|\text{transition}) = [p(1|1)\; p(0|0) + p(1|0)\; p(0|1)] \tag{4}$$

and

$$P_f = p(\text{transition}|\text{no transition}) = [p(1|0)\; p(0|0) + p(1|1)\; p(0|1)] \quad . \tag{5}$$

It is interesting to note that the expression in Equation (4) is independent of the prior probabilities of the deviation pairs (i,j), whereas Equation (5) requires that all variance pairs have equal probability,

$$p(1,0) = p(0,1) = p(0,0) = p(1,1) = 1/4 \quad . \tag{6}$$

This condition was maintained in Monte Carlo simulations which are described in following sections. The conditional probabilities appearing in Equations (4) and (5) are given by

$$p(1|i) = \int_{\gamma}^{\infty} p_i(x)\, dx \tag{7}$$

and

$$p(0|i) = \int_{o}^{\infty} p_i(x)\, dx \quad , \tag{8}$$

where p_i is the function p(x) in Equation (2) with σ given by σ_0 or σ_1 for i of 0 or 1, respectively.

The relationship between P_f and P_d as the threshold γ is varied, known as the receiver operating characteristic (ROC), characterizes the hypothesis test. Equations (4) and (5) refer to a decision on the existence of a variance transition in a Gaussian random process. The situation of no transition is distinguished from high/low or low/high variance transition. The similarity of the test to the binary exclusive-or map suggests the identification as the stochastic exclusive-or test.

As revealed in Equations (4) to (8), detection and false alarm probabilities for classical transition detection depend upon the threshold parameter γ. Figures 2(a-d) to 4(a-d) contain plots of false alarm and detection probability as a function of threshold for noise deviations σ_0 of one and σ_1 of two, four, and eight. As discussed in Sections III and IV, these are the deviations upon which neural net performance was evaluated. The figures within each set correspond to different widths N of the sampling window. The plots were constructed by numerical computation of P_f and P_d for each γ from Equations (4) to (8).

For each pair of deviations (σ_0,σ_1) and each sampling window N, the maximum of detection and minimum of false alarm probability occur at intermediate values of threshold parameter. At large values of $\sqrt{N}\,(\sigma_1 - \sigma_0)$ the optimum threshold was not obtained numerically within the figure domain [see Figures 4(a-d)]. Figure 5 contains the false alarm probability curves for fixed window (N = 10) as σ_1 is varied. Note that the optimum threshold slowly increases with the value σ_1 for fixed window widths. The optimum threshold is obtained from the condition

$$\frac{dP_f}{d\gamma} = 0 \quad , \tag{9}$$

which from Equations (4) to (8) is expressed as

$$p_1(\gamma)\left[1/2 - 2\int_{o}^{\gamma} p_0(x)\, dx\right] + p_0(\gamma)\left[1/2 - 2\int_{o}^{\gamma} p_1(x)\, dx\right] = 0 \quad . \tag{10}$$

Although Equation (10) can be readily solved by Newton's method, values of the optimum threshold (and corresponding P_f and P_d) were obtained directly from the plots in Figures 2(a-d) to 4(a-d). This provided sufficient accuracy for the comparison with neural net performance, as summarized in following sections.

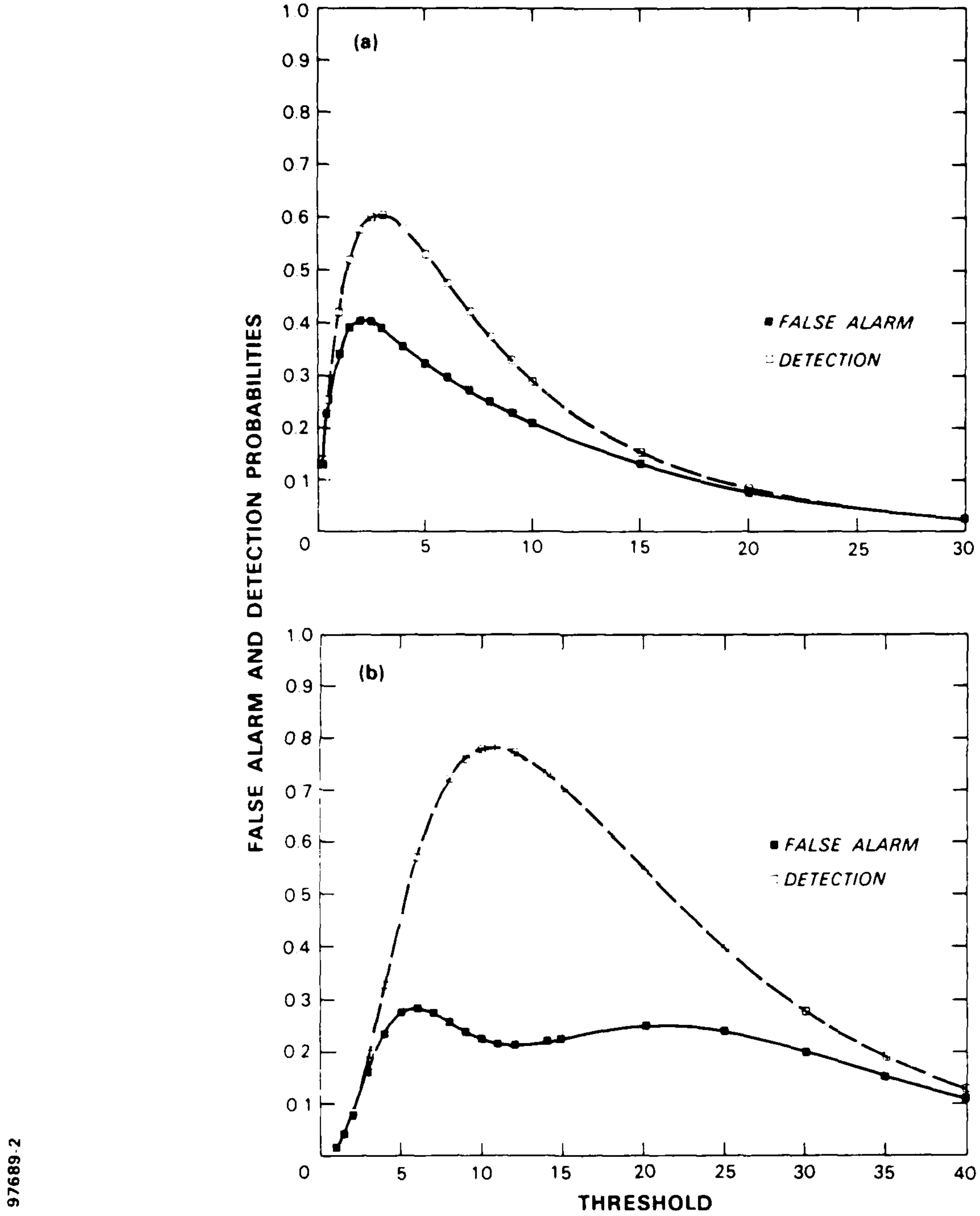


Figure 2. *False alarm and detection probability vs threshold parameters; noise deviations $\sigma_0 = 1.0$ and $\sigma_1 = 2.0$. (a) 2-sample window, (b) 6-sample window, (c) 10-sample window, and (d) 20-sample window.*

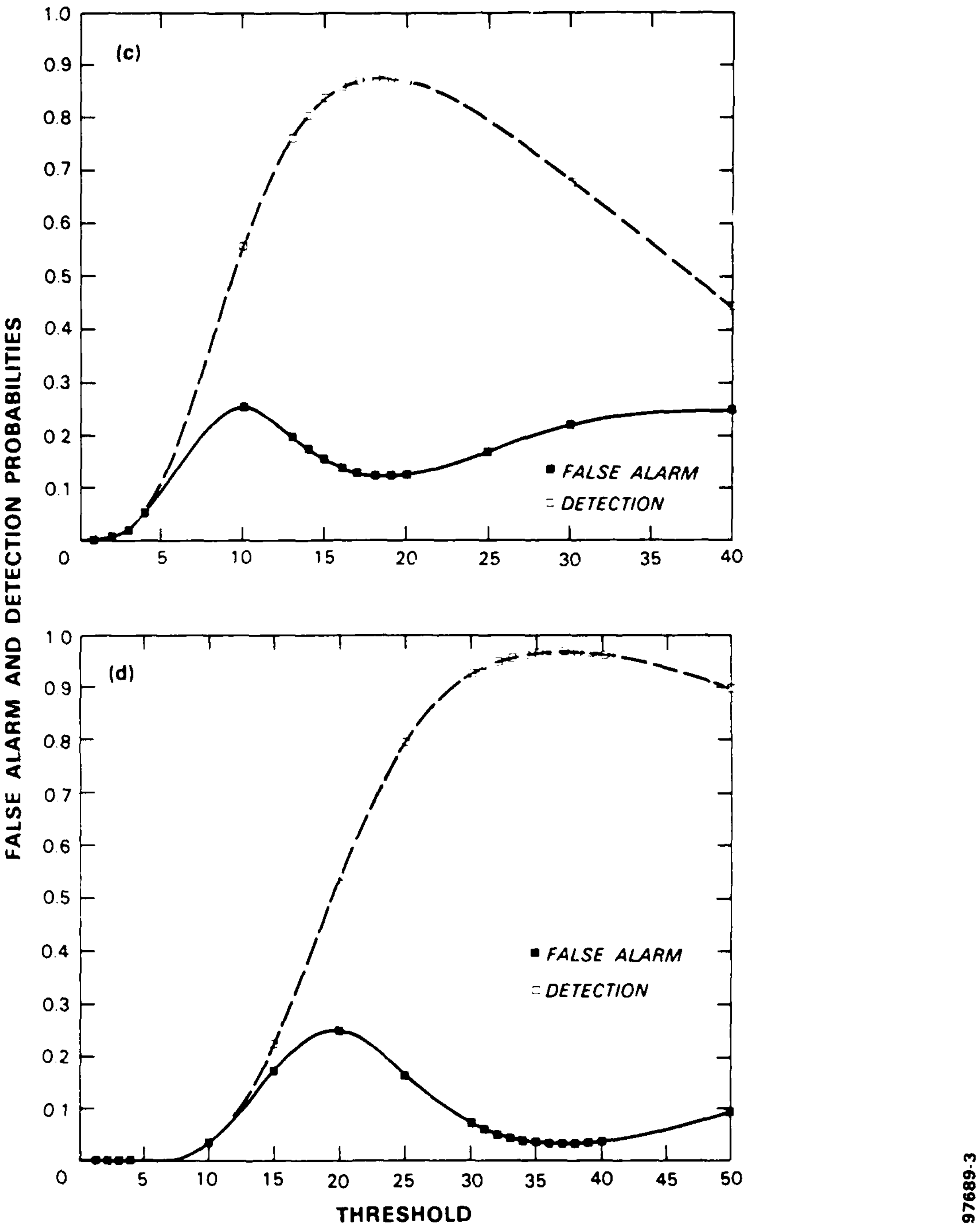


Figure 2. Continued.

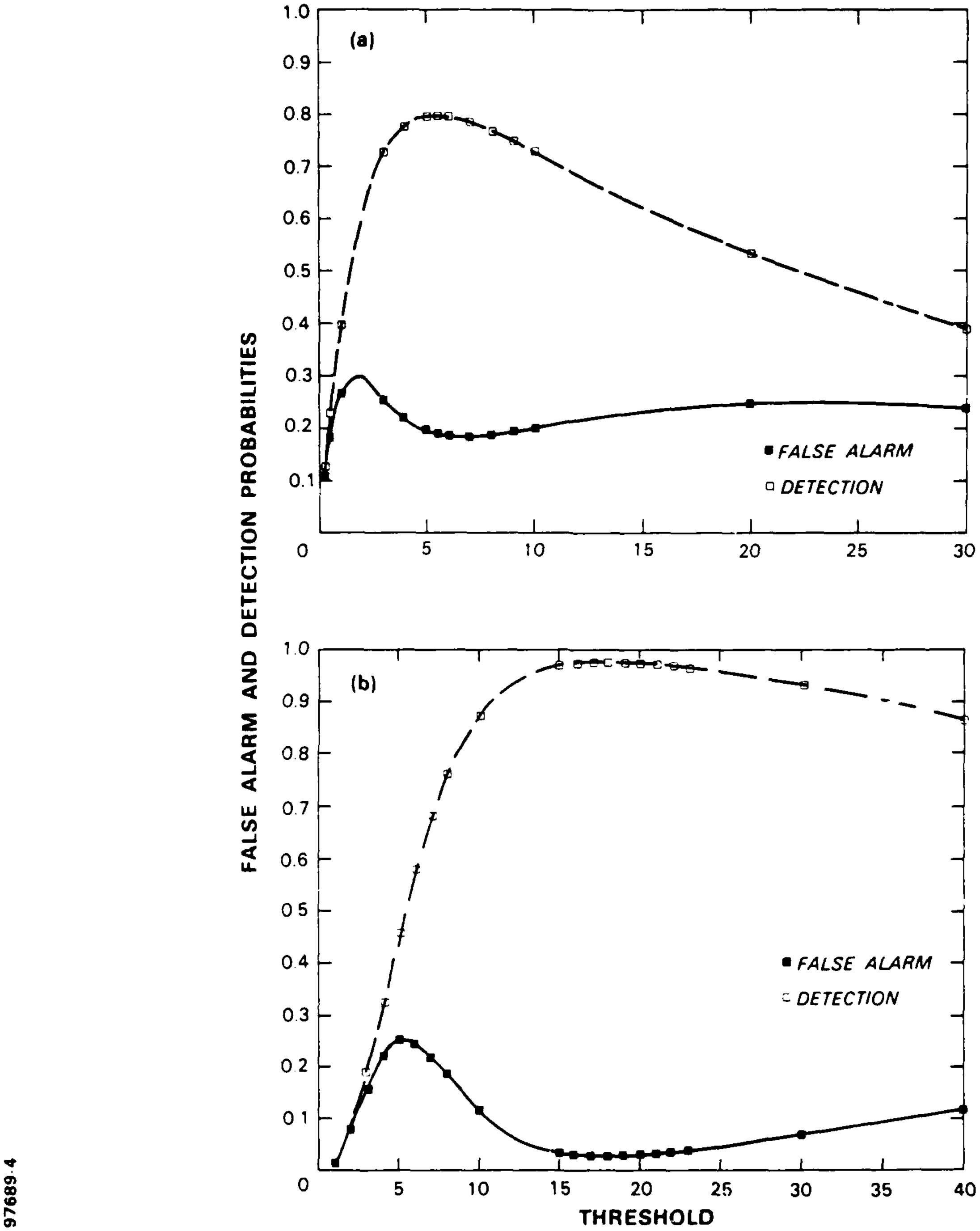


Figure 3. False alarm and detection probability vs threshold parameter; noise deviations $\sigma_0 = 1.0$ and $\sigma_1 = 4.0$. (a) 2-sample window, (b) 6-sample window, (c) 10-sample window, and (d) 20-sample window.

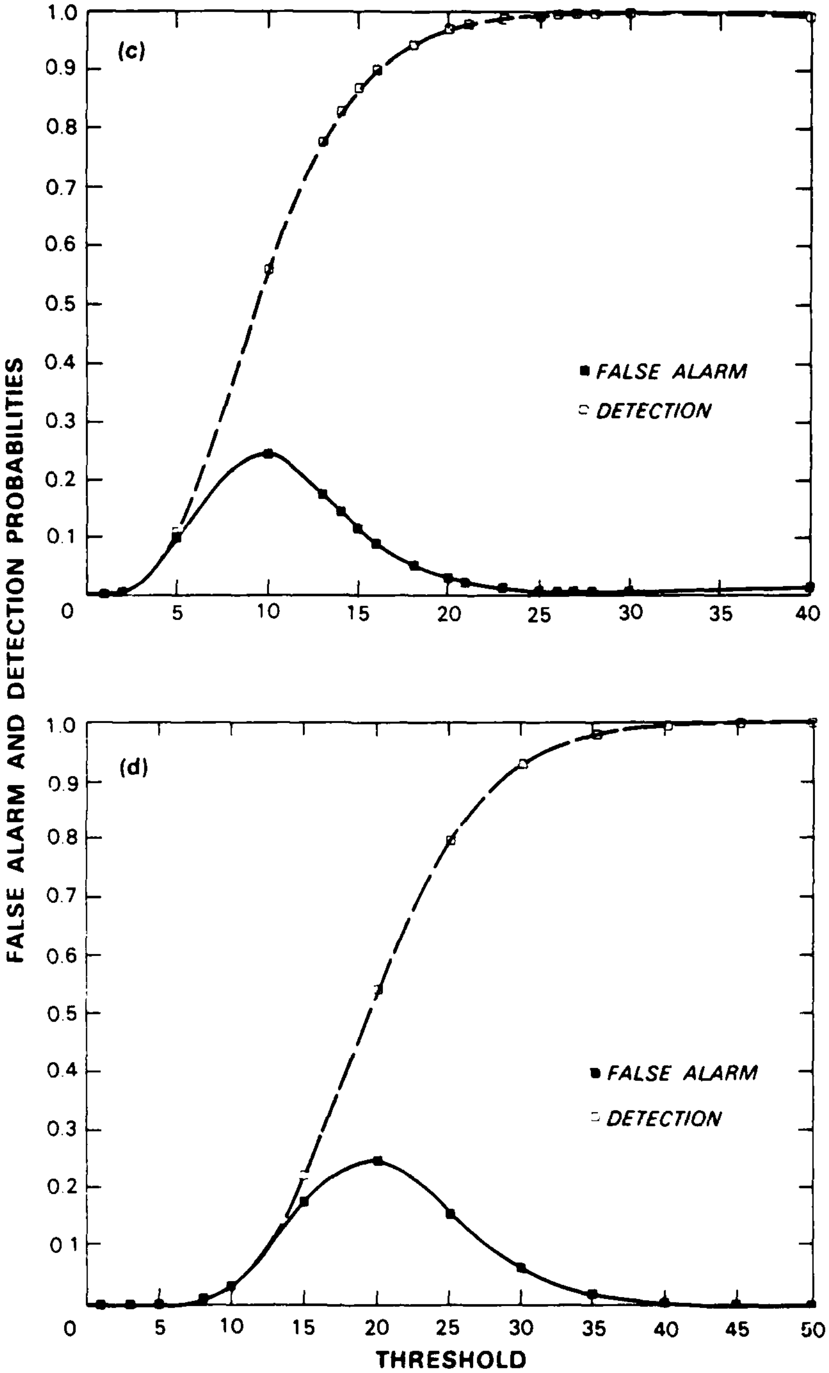
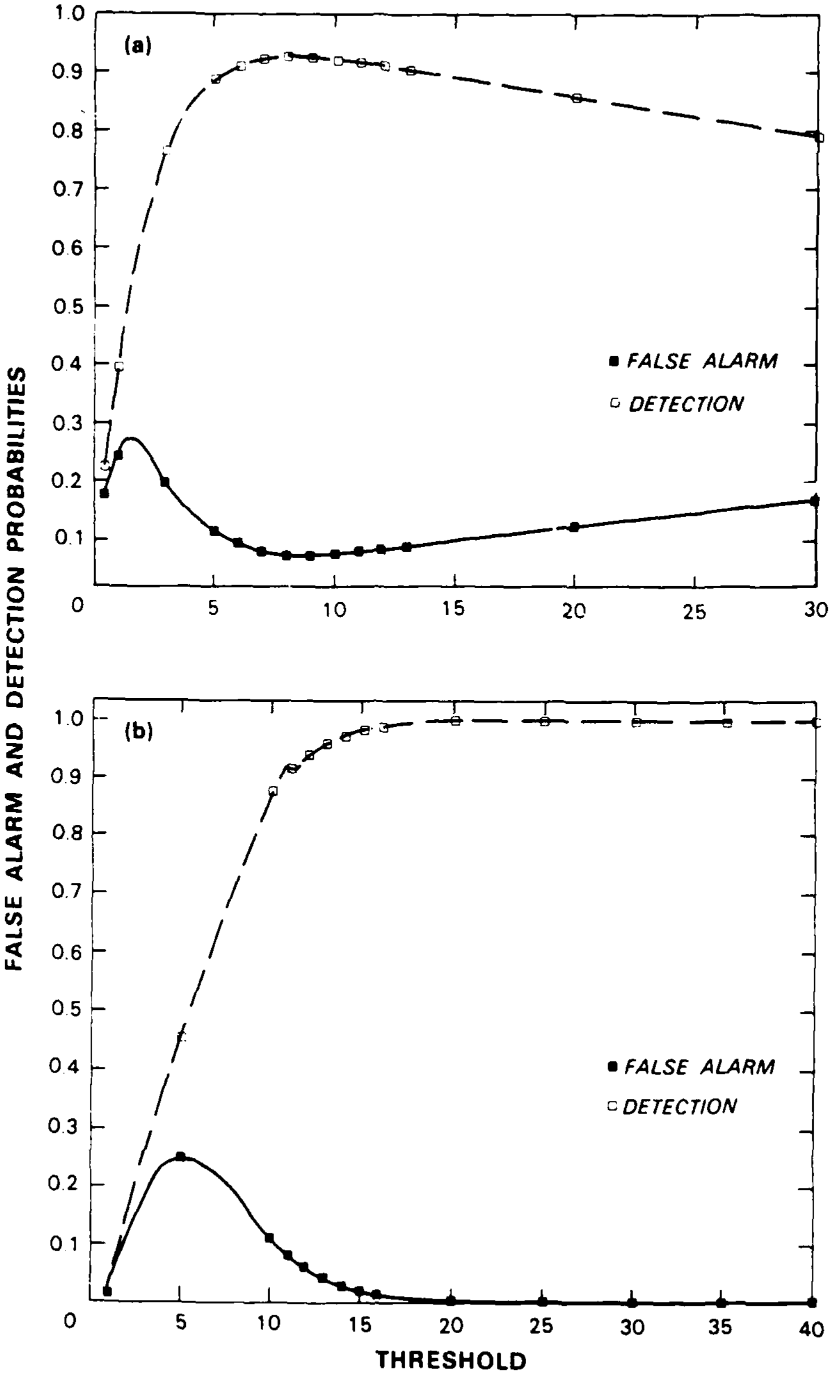


Figure 3. Continued.



97689-6

Figure 4. False alarm and detection probability vs threshold parameter; noise deviations $\sigma_0 = 1.0$ and $\sigma_1 = 8.0$. (a) 2-sample window, (b) 6-sample window, (c) 10-sample window, and (d) 20-sample window.

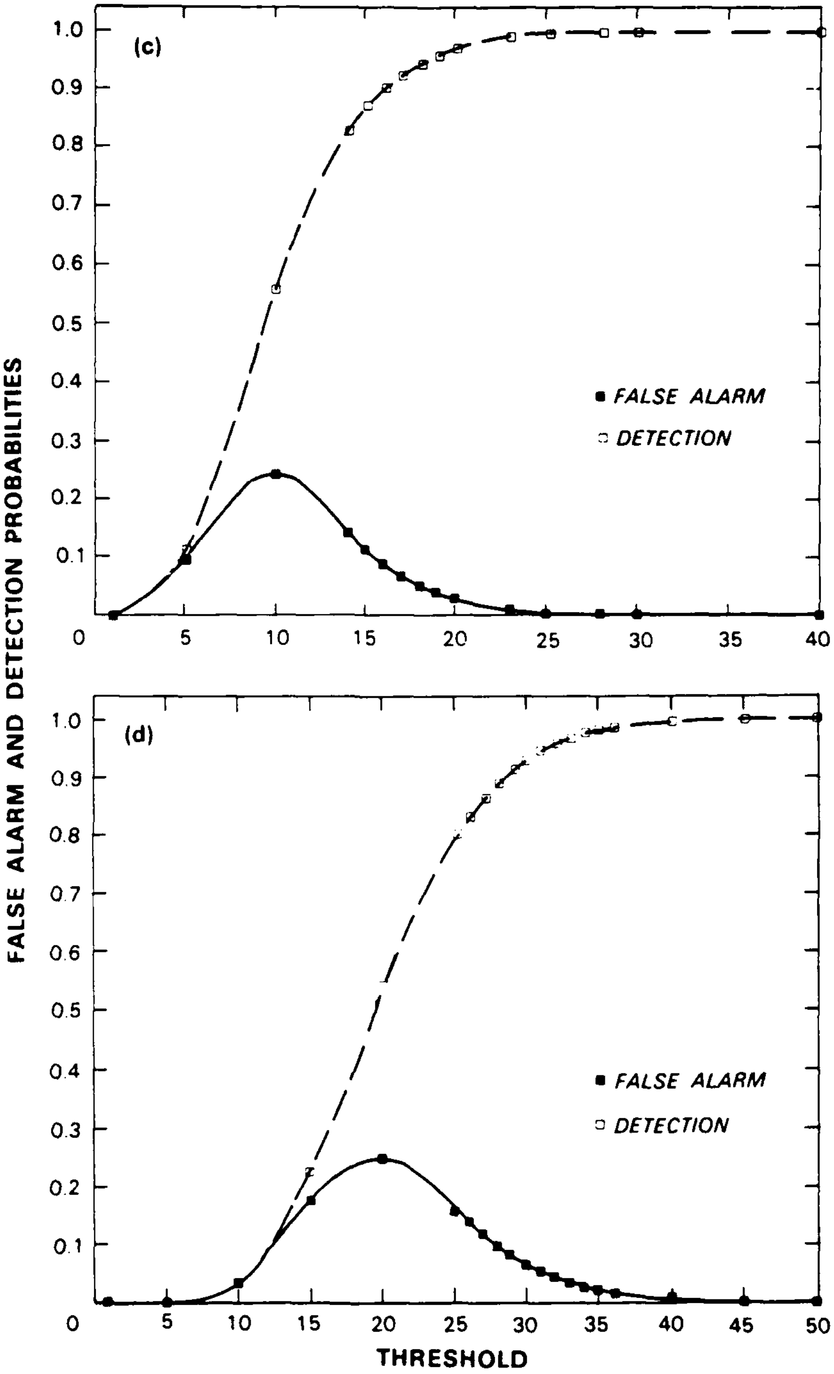


Figure 4. Continued.

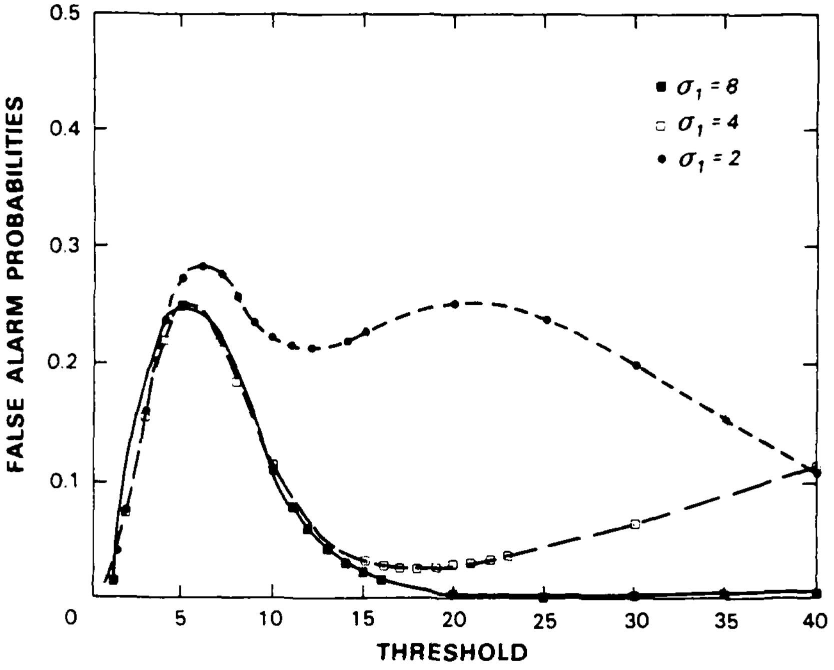


Figure 5. False alarm and detection probability vs threshold parameter for 6-sample windows. Cases of σ_l = 2, 4, and 8 considered separately.

III. MONTE CARLO TEST OF THE PERCEPTRON

In this section the performance of a perceptron neural network on the stochastic exclusive-or problem is analyzed. False alarm and detection probabilities as a function of training iteration are computed by Monte Carlo simulation. Figure 6 contains the minimal perceptron net required to perform the stochastic exclusive-or mapping.[7] Sample variances (χ_1, χ_2) computed from a pair of nonoverlapping data windows form the net inputs. The initial operation of the net in Figure 6 is the computation of

$$A_0 + A_1\chi_1 + A_2\chi_2 + A_1\chi_1\chi_2 \quad , \tag{11}$$

where (A_0,A_1,A_2,A_3) is the connection vector. A transition (or no transition) is implied if the quantity in Equation (11) is positive (or negative). Variances are computed on fixed length windows taken before and after the point in which the transition may have occurred. The variances form a pair of χ^2-distributed random variables with mean value $\sqrt{N}\sigma$, where N is the window length and σ the standard deviation of the Gaussian process. A schematic representation of the hypothesis test as a mapping is shown in Figure 7.

Connection weights (A_1,A_2,A_3) and the threshold A_0 are adjusted from initially random values by application of a training ensemble of input variance pairs (for which the existence of a transition is known). The training algorithm is exactly that which appears in the perceptron convergence theorem of Reference 7. A normalized predicate vector $\hat{\Phi}$ is defined from the input pair (χ_1,χ_2) as,

$$(1,\chi_1,\chi_2,\chi_1\chi_2)/\sqrt{1 + \chi_1^2 + \chi_2^2 + (\chi_1\chi_2)^2} \quad . \tag{12}$$

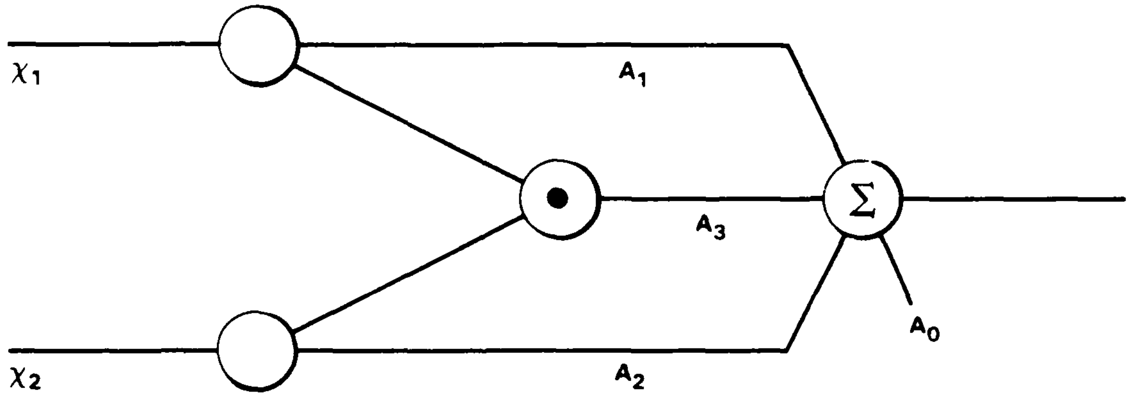


Figure 6. A minimal neural net structure for transition detection based upon the exclusive-or logic element.

Starting with a random connection weight-threshold four-vector,

$$A = (A_0, A_1, A_2, A_3) \tag{13}$$

alter A if

(1) $\hat{\Phi}$ corresponds to a transition and $A \cdot \hat{\Phi} < 0$,

(2) $\hat{\Phi}$ corresponds to no transition and $A \cdot \hat{\Phi} > 0$,

In either case adjust A additively with the normalized predicate vector,

$$A^{(i)} = A^{(i-1)} \pm \hat{\Phi} \quad , \tag{14}$$

where addition and subtraction correspond to cases (1) and (2), respectively. The effect of Equation (14) is the adjustment of the product $A \cdot \hat{\Phi}$ positively for a transition and negatively for no transition.

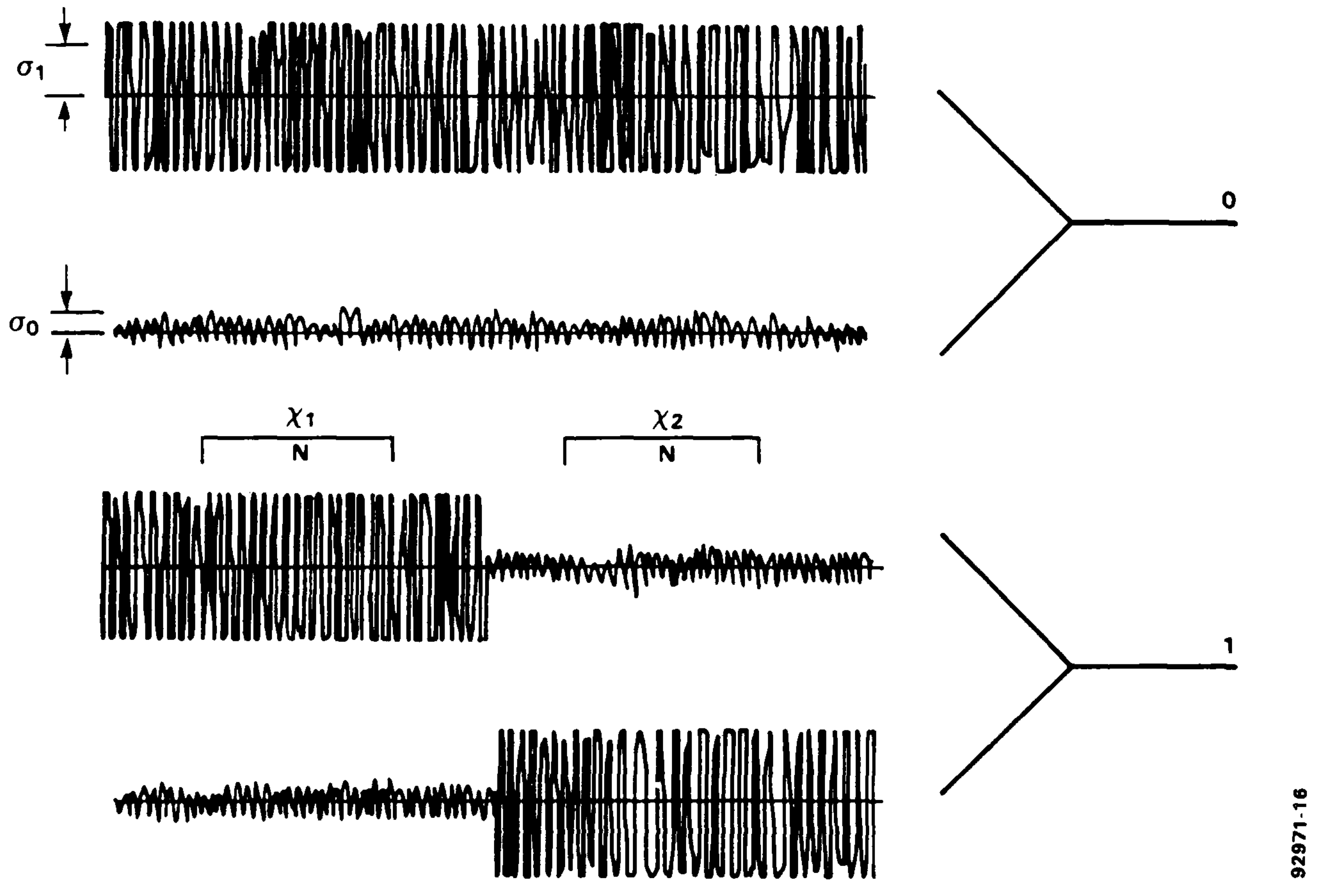


Figure 7. Schematic representation of stochastic exclusive-or map for variance transition.

The use of the network in Figure 6, with the corresponding four-vector A, is motivated by the linear separation principle of perceptron learning.[7] Consider a two-dimensional space consisting of a before and after variance pair (χ_1, χ_2) from an ensemble of stochastic data. As shown in Figure 8, the task of transition detection is to differentiate a region of probability clouds centered at $(\sqrt{N}\sigma_0, \sqrt{N}\sigma_0)$ and $(\sqrt{N}\sigma_1, \sqrt{N}\sigma_1)$ from probability clouds at $(\sqrt{N}\sigma_0, \sqrt{N}\sigma_1)$ and $(\sqrt{N}\sigma_1, \sqrt{N}\sigma_0)$. As shown in Figure 9, the embedding,

$$(\chi_1, \chi_2) \longrightarrow (\chi_1, \chi_2, \chi_1\chi_2) \tag{15}$$

enhances the linear separability of these regions, thereby suggesting the structure in Figure 6.

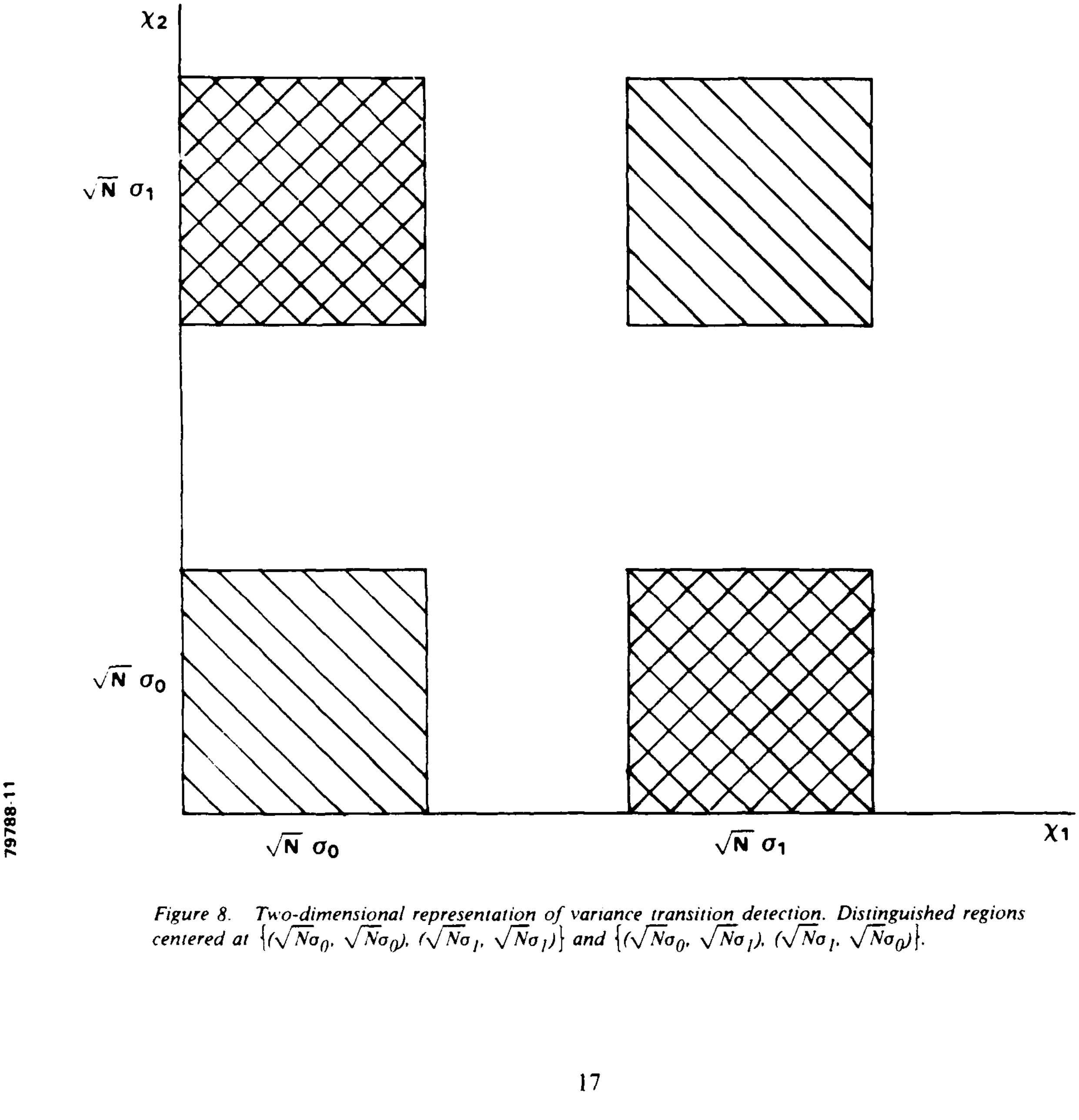


Figure 8. Two-dimensional representation of variance transition detection. Distinguished regions centered at $\{(\sqrt{N}\sigma_0, \sqrt{N}\sigma_0), (\sqrt{N}\sigma_1, \sqrt{N}\sigma_1)\}$ *and* $\{(\sqrt{N}\sigma_0, \sqrt{N}\sigma_1), (\sqrt{N}\sigma_1, \sqrt{N}\sigma_0)\}$.

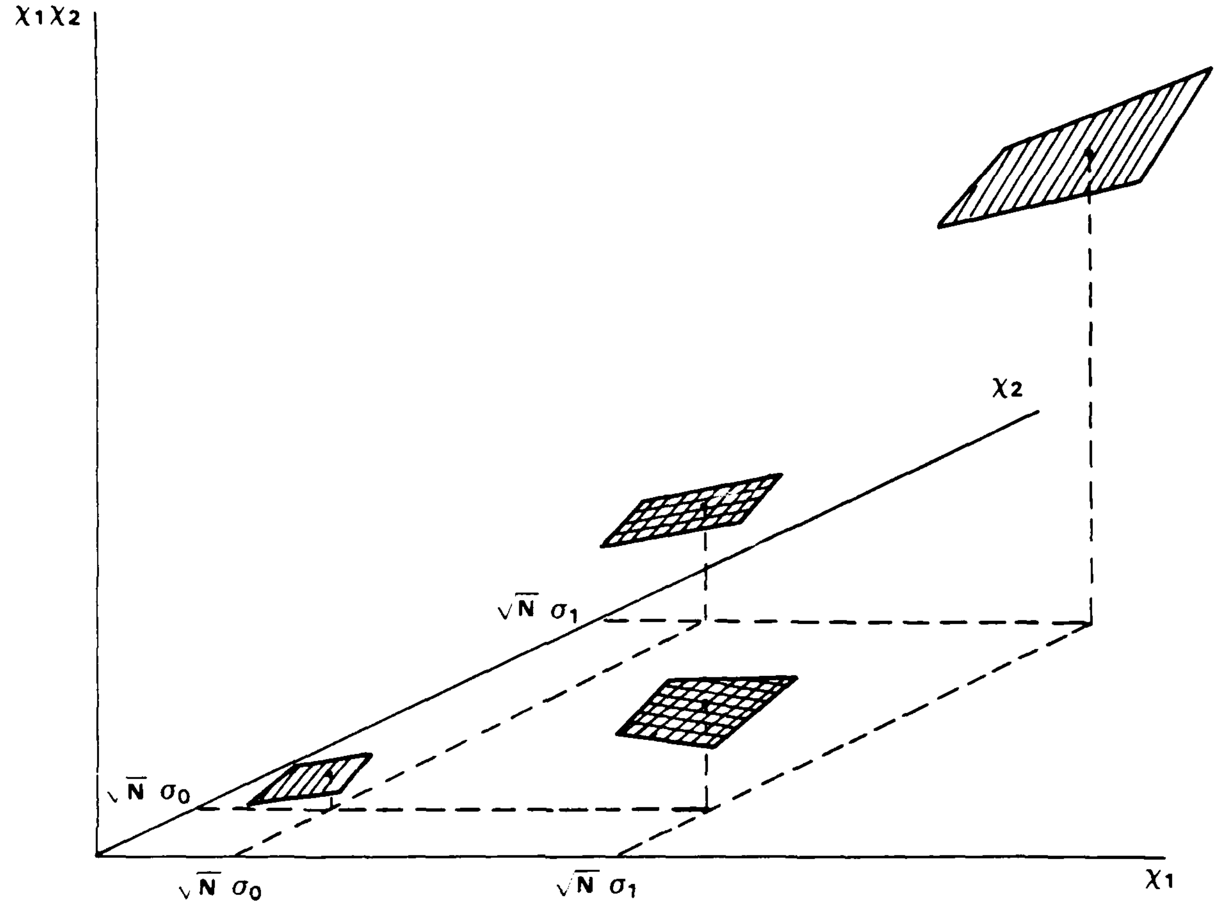


Figure 9. Embedding of transition detection map to three dimensions. Enhanced linear separability of regions in Figure 8.

In order to model perceptron performance on stochastic data, a training ensemble of NT window pairs was created. Each window contained N samples of Gaussian random noise with standard deviation σ_0 or σ_1. The training algorithm was applied to a set of variance pairs $[\chi_1^{(p)}, \chi_2^{(p)}]$ which as a function of p sequence through possible noise conditions (σ_1,σ_0), (σ_1,σ_1), (σ_0,σ_1), and (σ_0,σ_0). The range of the net mapping corresponding to this sequence alternates between transition and no transition hypotheses. Upon completion of a number of training iterations a performance ensemble was created which contained 500 transition and no transition trace pairs. Ensemble-derived detection and false alarm probabilities were obtained simply by counting the number of detected transitions and falsely detected no transitions in the ensemble. Detection and false alarm probabilities are functions of the number of iterations of the training algorithm.

Figures 10(a-d), 11(a-d), and 12(a-d) contain false alarm and detection probability vs NT for standard deviation σ_0 of one and σ_1 the two, four, and eight, respectively. Within each figure set

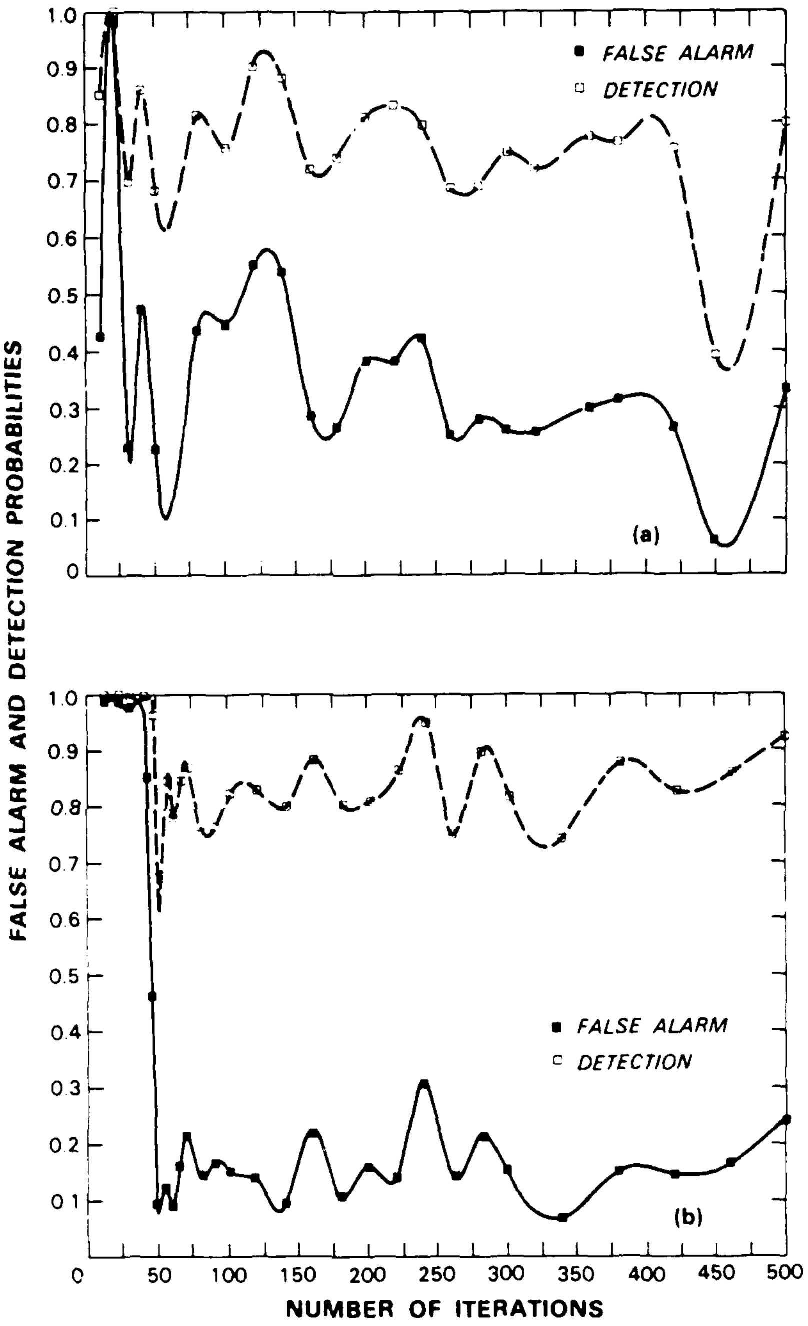


Figure 10. False alarm and detection probability vs iteration number (NT); noise deviations $\sigma_0 = 1.0$ and $\sigma_1 = 2.0$. (a) 5-sample window, (b) 10-sample window, (c) 20-sample window, and (d) 40-sample window.

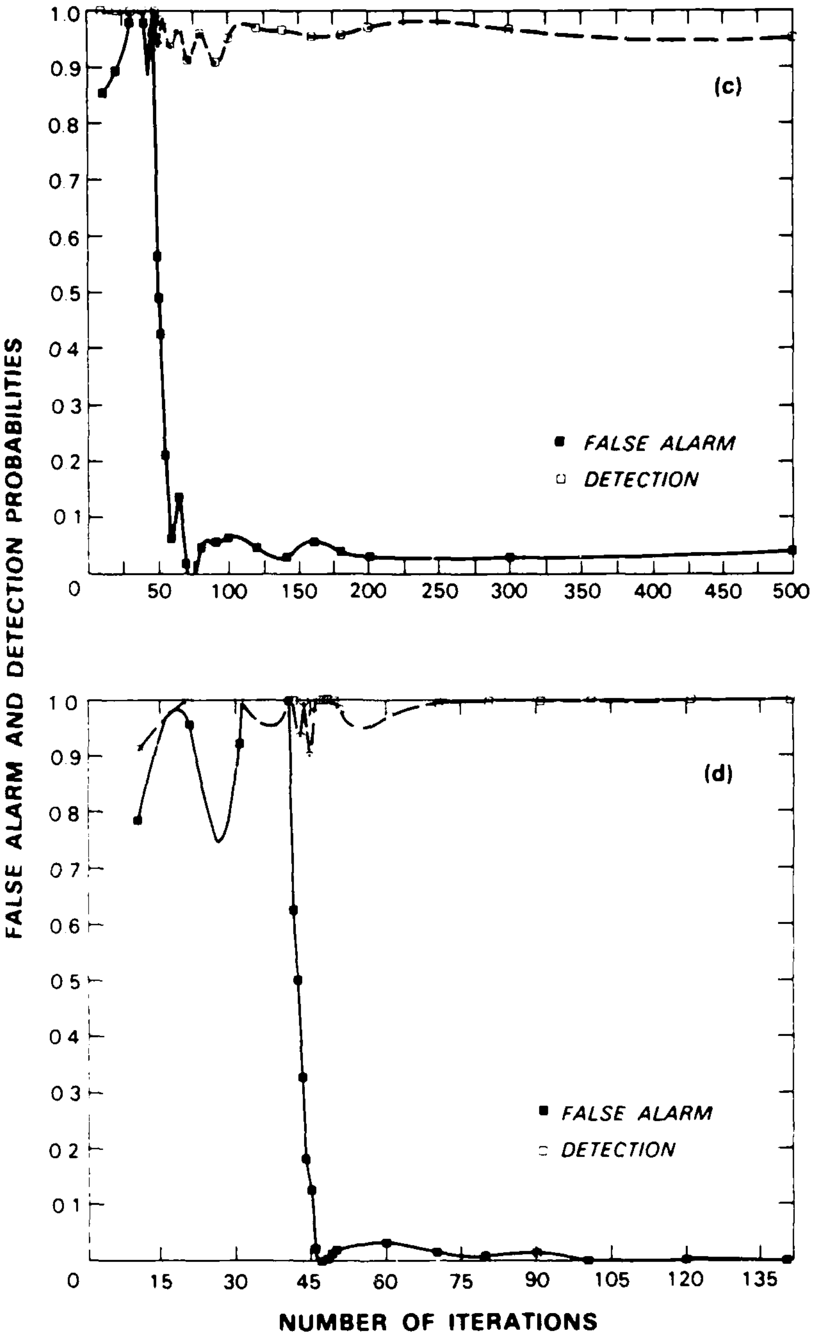


Figure 10. Continued.

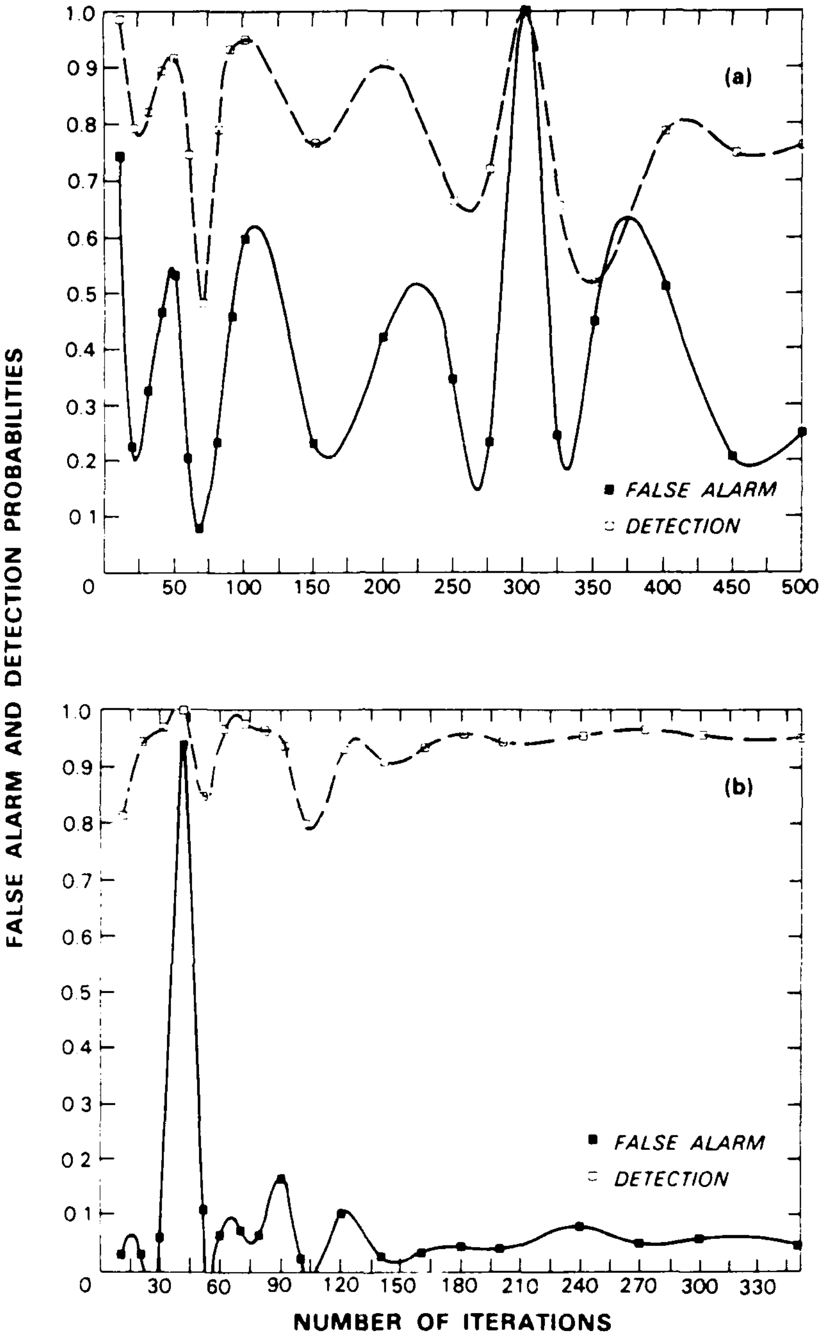
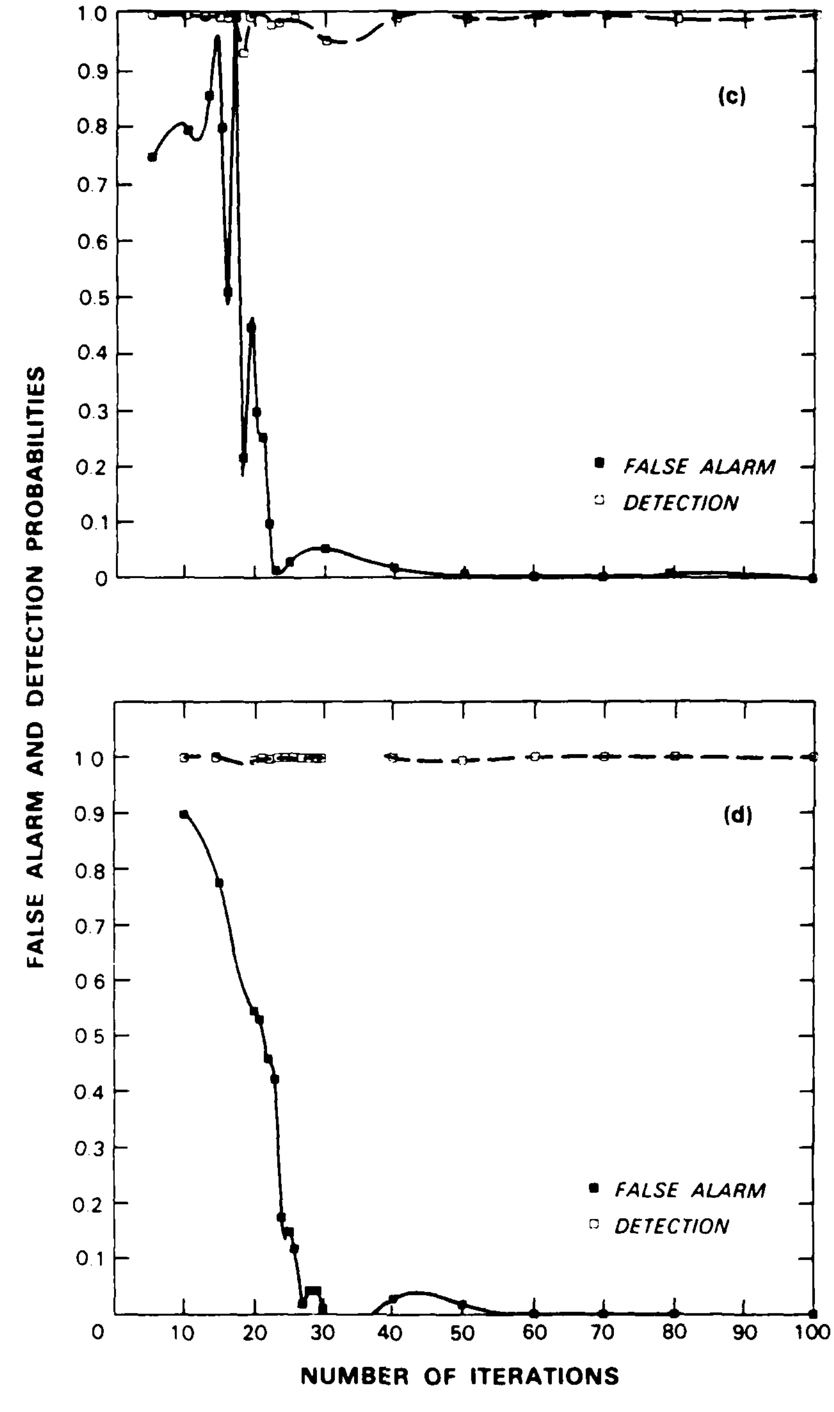


Figure 11. False alarm and detection probability vs iteration number (NT); noise deviations $\sigma_0 = 1.0$ and $\sigma_1 = 4.0$. (a) 2-sample window, (b) 5-sample window, (c) 10-sample window, and (d) 20-sample window.



97689-13

Figure 11. Continued.

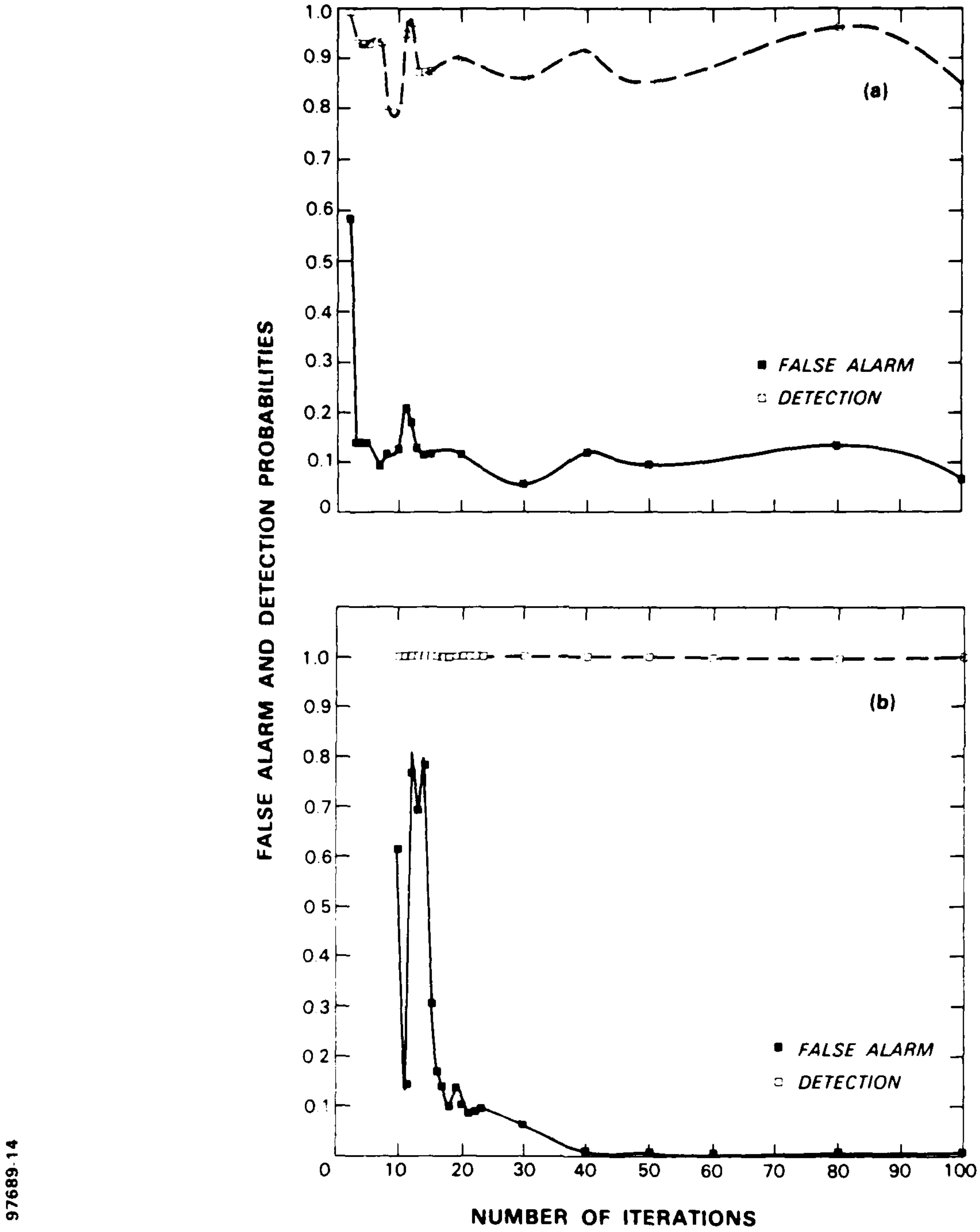
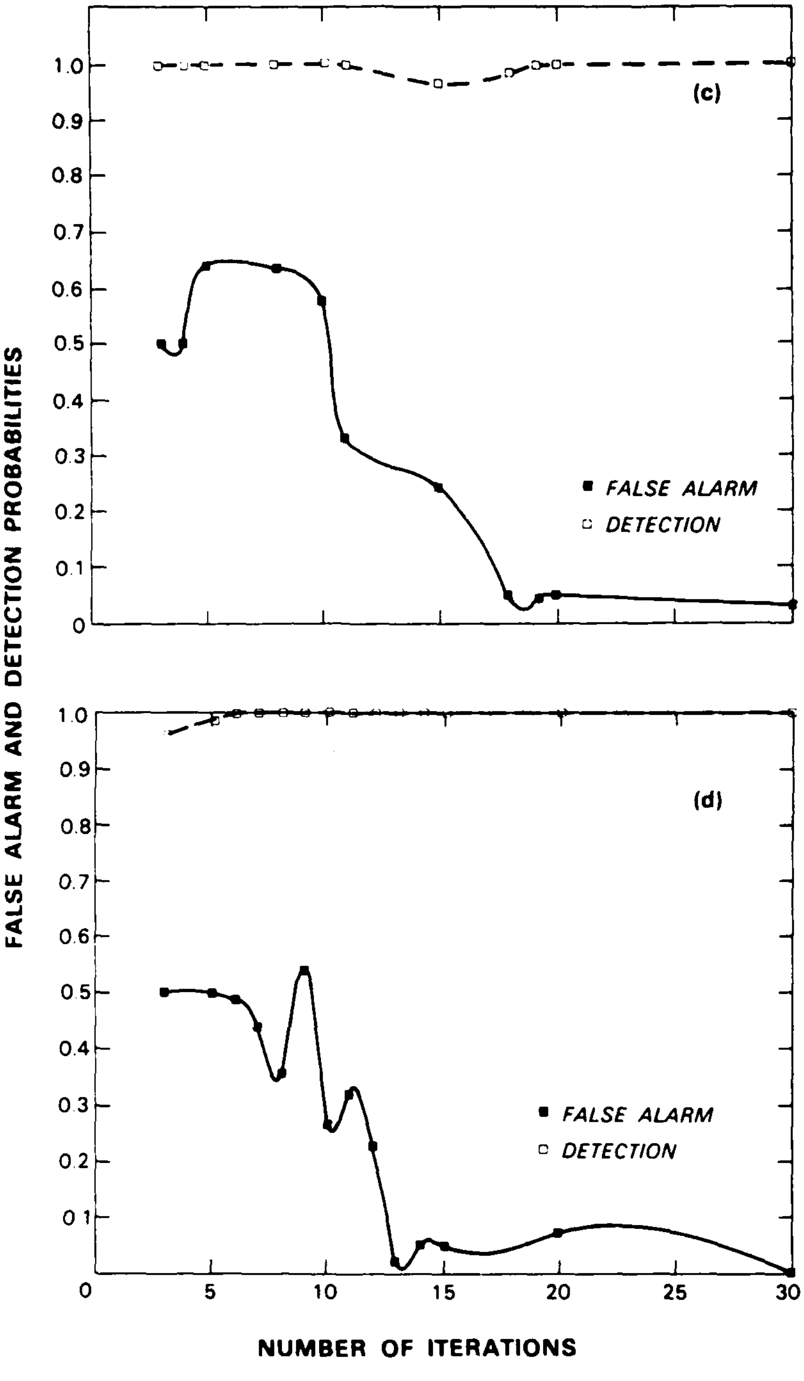


Figure 12. False alarm and detection probability vs iteration number (NT); noise deviations $\sigma_0 = 1.0$ and $\sigma_1 = 8.0$. (a) 2-sample window, (b) 5-sample window, (c) 10-sample window, and (d) 20-sample window.



97689-15

Figure 12. Continued.

the length of the windows is varied such that (a) to (d) correspond to increasing sample number N. Note that in all cases detection and false alarm probabilities were initially near one due to the randomly chosen initial connection weight four-vector $A^{(0)}$ [Equation (13)]. Learning is revealed by the sudden drop in false alarm probability; a characteristic which reflects the nonlinear nature of the training algorithm. The nonlinearity appears in the product $\chi_1\chi_2$ of the predicate $\hat{\Phi}$ [Equation (12)].

As seen in Figures 10(b), 11(b), and 12(a), unambiguous transition detection for σ_1 of two, four, and eight required window lengths of ten, five, and two, respectively. Figure 13 contains the false alarm probability vs training iteration for 10 sample windows as σ_1 is varied. Note that generally learning is observed in <100 iterations with those combinations of σ_1 and N for which net connections converge. Asymptotic false alarm and detection probability (in the limit of many iterations) correspond to the optimum performance of the perceptron. These values are compared with classical test performance at optimum thresholds in Section V.

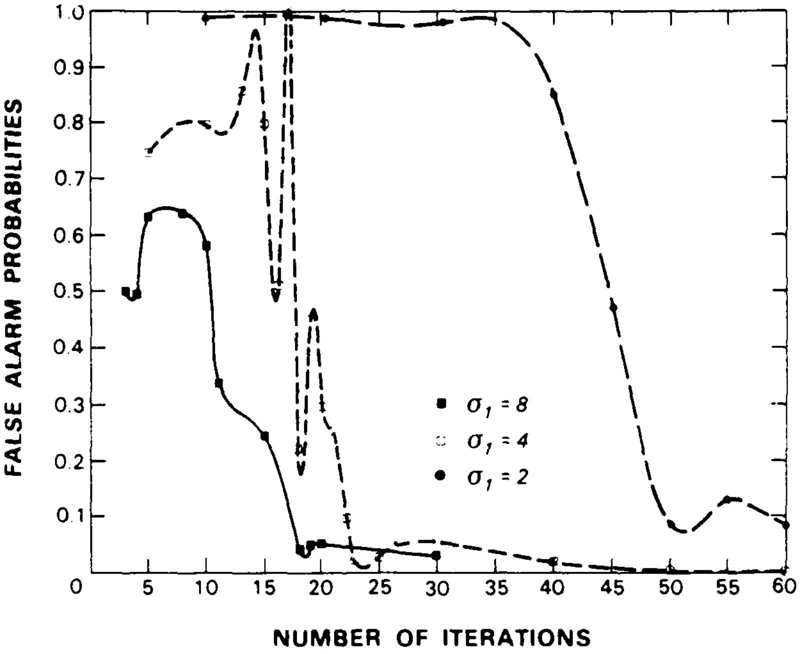


Figure 13. False alarm probability vs iteration number (NT) for 10-sample windows. Values σ_I of 2, 4, and 8 considered separately.

IV. MONTE CARLO TEST OF BACK PROPAGATION

Back propagation networks are a recently discovered modification of the perceptron which involve only first-order neural units.[2] Figure 14 contains a back propagation network appropriate for the stochastic exclusive-or mapping. It consists of two inputs, *H middle* layer neurons, and a single output neuron. Network parameters consist of thresholds $\{\theta_i | i = 1,2,...,2H + 4\}$ at each neuron and connection weights $\{W_{ij} | i,j = 1,2,...,2H + 4\}$ between neuron pairs. The operation at the (i,j) connection is to multiply the output of the leftmost neuron (j) of the pair by W_{ij} before input to the rightmost neuron (i). The total input to a neuron i is then

$$I_i = \sum_{j \epsilon \Xi} W_{ij} O_j \quad , \tag{16}$$

where Ξ is the set of neurons connected to i from the left, and O_j is the output of neuron j. The output of neuron i is given by

$$O_i = \frac{1}{\{1 + \exp[-I_i + \theta_i]\}} \quad , \tag{17}$$

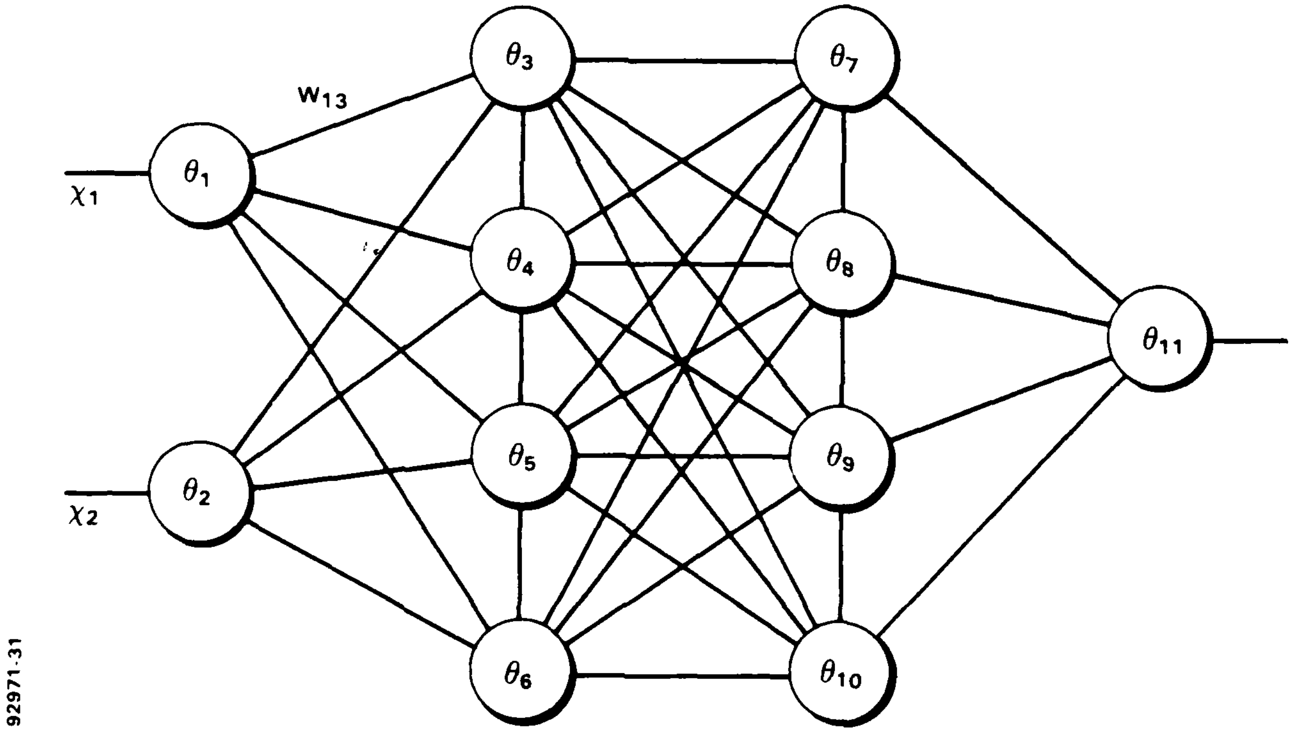


Figure 14. Back propagation network for stochastic exclusive-or two-input neurons, H hidden layer neurons, single output. Connection weights $\{W_{ij} | i,j = 1,2,...,H + 4\}$ thresholds $\{\theta_i | i = 1,2,...,H + 4\}$.

corresponding to nonlinear threshold activation at θ_j. The nonlinear activation function in Equation (17) is roughly analogous to the threshold which is applied upon output of the perceptron (see Figure 6).

The adaption of connection weights and thresholds, resulting in a match between output and target, is implemented by the back propagation learning algorithm. The derivation of the algorithm, which is based upon a least-squares minimization, and a discussion of numerous implementation issues, can be obtained in Reference 2. For the purpose of this report, an outline of the algorithm will be presented in order to define required auxiliary parameters. A training ensemble is defined $\{[\chi_1^{(p)}, \chi_2^{(p)}, t^{(p)}] | p = 1,2,...,NT\}$, consisting of input variance pairs $[\chi_1^{(p)}, \chi_2^{(p)}]$ and an output transition flag $t^{(p)}$. The parameter $t^{(p)}$ is one (zero) if $\{[\chi_1^{(p)}, \chi_2^{(p)}]\}$ corresponds to a transition (no transition) condition. The learning algorithm is defined recursively starting with a randomly chosen set of initial connection weights $\{W_{ij}^{(0)}\}$ and thresholds $\{\theta_j^{(0)}\}$. Assuming that training on $\{[\chi_1^{(1)}, \chi_2^{(1)}], ..., [\chi_1^{(p-1)}, \chi_2^{(p-1)}]\}$ has resulted in net parameters $\{W_{ij}^{(p-1)}\}$ and $\{\theta_j^{(p-1)}\}$, the input of pair $[\chi_1^{(p)}, \chi_2^{(p)}]$ results in output $O^{(p)}$. Define the output error

$$\delta^{(p)} = [t^{(p)} - O^{(p)}] \quad , \tag{18}$$

and backward propagate error to each neuron j by

$$\delta_j^{(p)} = O_j^{(p-1)} [1 - O_j^{(p-1)}] \sum_k \delta_k^{(p)} W_{jk}^{(p-1)} \quad , \tag{19}$$

where $O_j^{(p-1)}$ was the output of neuron j on the (p-1)st iteration. The connection weight W_{ij} between neurons i and j is altered by

$$\Delta W_{ij}^{(p)} = \eta\, \delta_j^{(p)} O_i^{(p-1)} + \alpha\, \Delta W_{ij}^{(p-1)} \quad , \tag{20}$$

where η controls the rate of connection weight modification and α is a smoothing parameter.[2] The threshold $\theta_j^{(p-1)}$ for the jth neuron is adjusted on the pth iteration by

$$\Delta\theta_j^{(p)} = \eta\, \delta_j^{(p)} \quad . \tag{21}$$

Note that the back propagation learning algorithm, Equations (19) to (21), requires the introduction of new parameters η and α. During training of the network in Figure 14 on stochastic data, the parameters were chosen to optimize performance, as measured by training time and smoothness of convergence.

Progress of the training procedure is measured by the match of output $O^{(p)}$ to target $t^{(p)}$ for each element of the training ensemble. This may be obtained through a Hamming-type of measure,

$$H = \sum_{p=1}^{NT} \theta[|t^{(p)} - O^{(p)}|] \tag{22}$$

where

$$\theta[|t^{(p)} - O^{(p)}|] = \begin{cases} 1 & |t^{(p)} - O^{(p)}| > \tau \\ 0 & |t^{(p)} - O^{(p)}| < \tau \end{cases} \qquad (23)$$

with a predetermined match parameter τ. Another training measure is simply the rms difference between the ensemble outputs and targets,

$$H = \sqrt{\sum_{p=1}^{NT} [O^{(p)} - t^{(p)}]^2} \quad . \qquad (24)$$

Figure 15 contains the Hamming and rms measures vs training iteration for a 40-element stochastic exclusive-or training ensemble. The network was that in Figure 14 with 64 hidden units. Noise deviations σ_0 and σ_1 were one and two, respectively; and the input windows contained 20 samples. The match parameter τ in Equation (23) was 0.1. The training algorithm was run repeatedly on the entire ensemble of 40 elements with alternating transition and no transition outputs. Note that after ~100 presentations of the ensemble smooth convergence of both Hamming and rms measures was interrupted by oscillations. The source of the training reversals, which appear commonly for some α parameters, has yet to be determined.

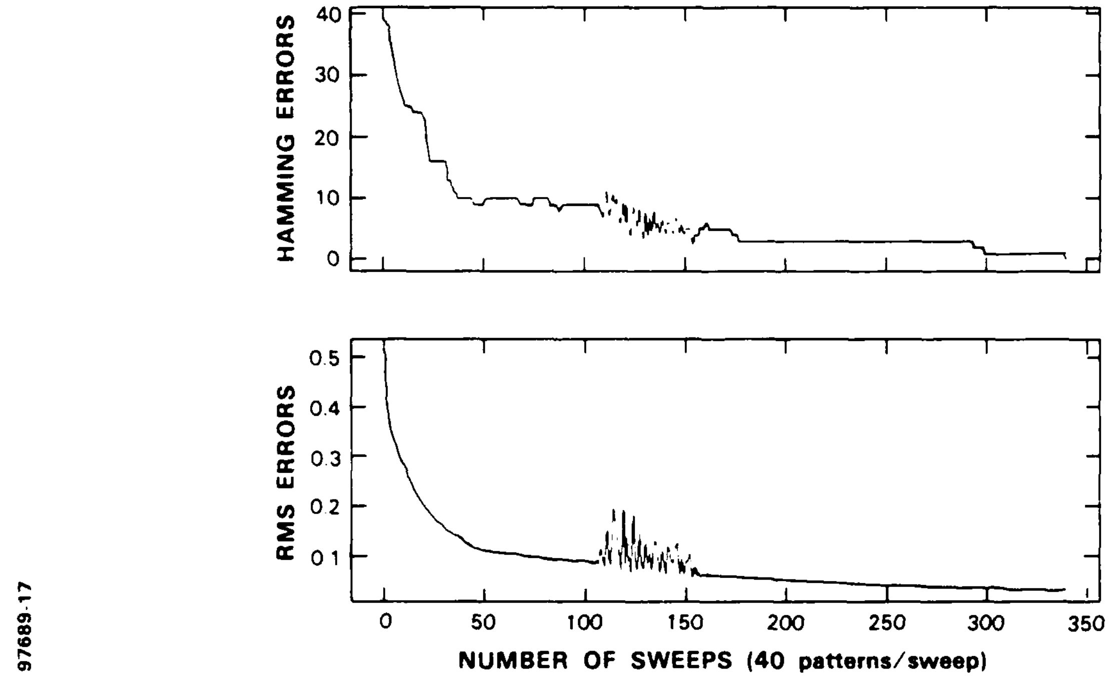


Figure 15. Hamming and rms measures vs iteration number (NT) for stochastic exclusive-or, $\sigma_0 = 1.0$, $\sigma_1 = 2.0$, $\alpha = 0.2$, $\eta = 0.3$, 64 hidden layer neurons, 20-sample window.

The performance of a trained back propagation network was measured by Monte Carlo simulation with the aid of a performance ensemble. This consisted of a set of NE variance pairs $[\chi_1^{(q)}, \chi_2^{(q)}]$ and the corresponding designator $t^{(q)}$ ($q = 1,2...,NE$). As in the case of perceptron modeling, false alarm and detection probabilities were computed as a function of iteration number (as implemented on the training ensemble) from net operation on the performance ensemble. A series of experimental runs determined that the optimum number H of hidden units is sixteen for the stochastic exclusive-or problem. In contrast to the binary output of a perceptron network, the output of the back propagation network is analog. This requires the addition of an output match parameter τ, which is the maximum allowed difference between the output and target for hypothesis determination.

Figures 16(a-c), 17(a-c), and 18(a-c) contain simulated P_f and P_d vs training iteration for windows of 5, 10, and 20 samples, respectively. In all cases the pair of noise deviations σ_0 and σ_1 were 1.0 and 2.0, respectively. In each of the simulations the smoothing parameter α was empirically chosen to optimize network performance. Within each figure set the P_f-P_d plots correspond to different sized training ensembles. For example, Figures 16(a), (b), and (c) contain false alarm and detection probability for input window lengths of 5 samples with training ensembles of 6, 10, and 40 elements. Note that for 5 sample windows the size of the training ensemble has little effect on network performance or training time. In the case of 10 sample windows, the use of a 40-element training ensemble halved the asymptotic false alarm probability over that of a 10-element ensemble. This comparison is obtained in Figures 17(a) and (c). Unlike the perceptron, back propagation networks converged on 5-sample windowed variances. However, the asymptotic (P_f, P_d) of about (0.6, 0.7) in Figure 16(c) is to be compared with the classical optimum of (0.2, 0.8) in Figure 2(b). The 10-sample variance inputs yield (P_f, P_d) of about (0.15, 0.9) with a 40-element training ensemble [Figure 17(c)], which is close to the classical optimum values in Figure 2(c).

A common feature of back propagation networks, as revealed in Figures 16 and 17, is the extraordinary number of training iterations (of order 10^4) required to obtain network convergence. A partial remedy is to scale the input training and performance ensemble variances to the range (0,1). The magnitudes of the required connection weight adjustments are smaller; untuitively resulting in faster convergence. Figures 18(a) to (c) contain a set of simulations on 20-sample variances, in which convergence time was reduced by a factor of 10 through input scaling. Note, however, that the asymptotic false alarm probability of 0.2 is four times the classical optimum of 0.05 in Figure 2(d). Because the mean value of an input variance is $\sqrt{N}\sigma$, it is expected that the scaling of inputs has the greatest effect for large N; a limit which corresponds to a large contraction of the map input domain.

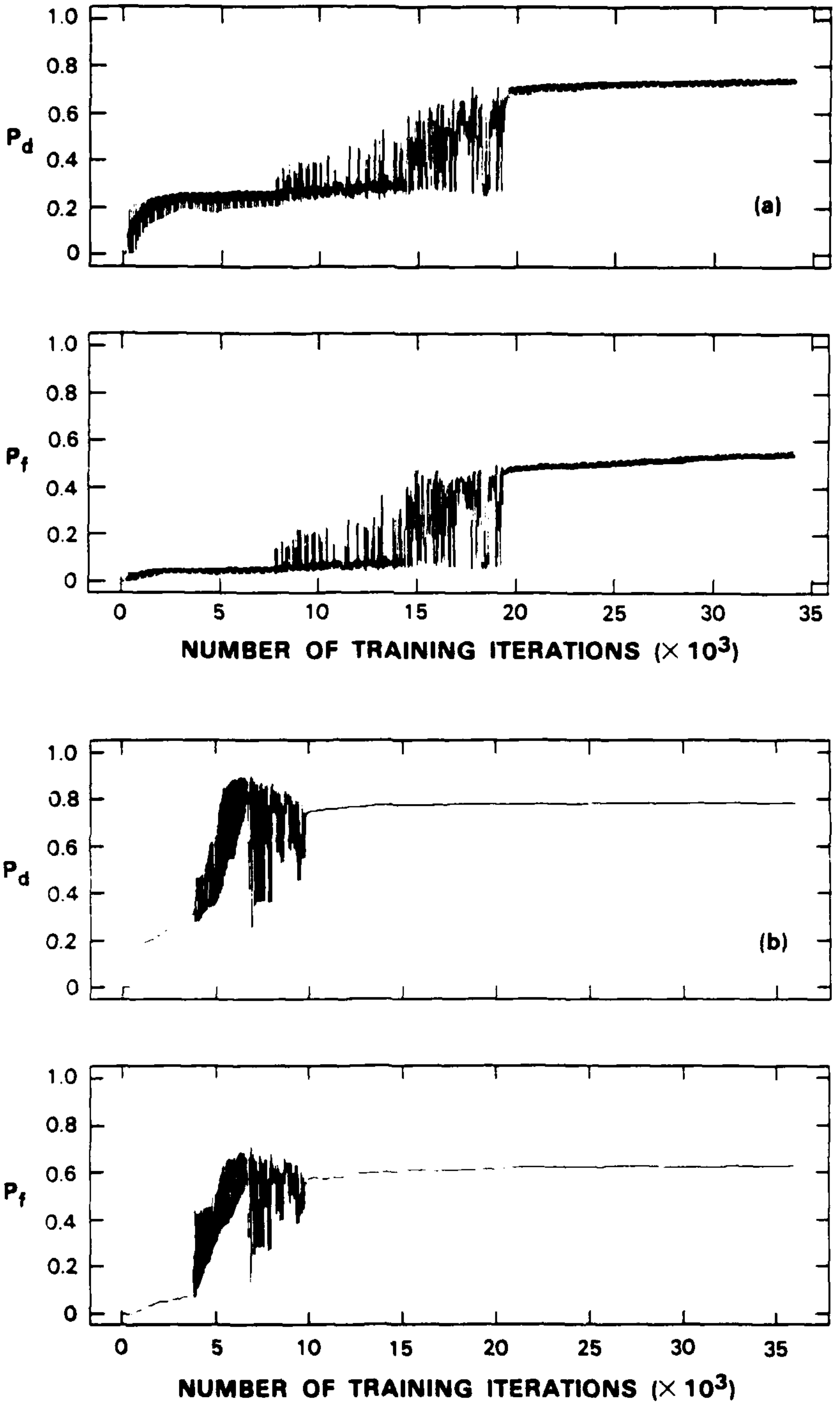
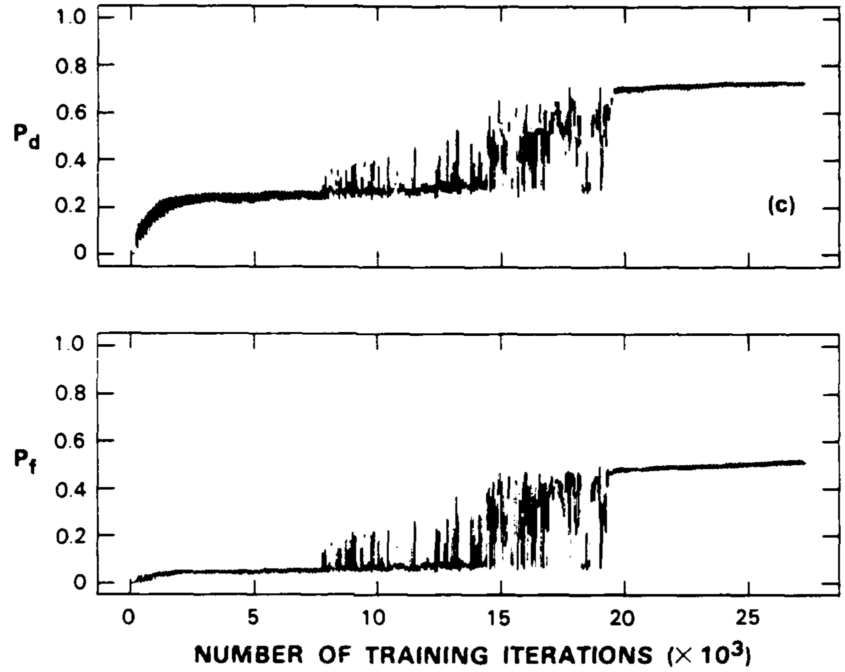


Figure 16. False alarm and detection probability vs iteration number; $\sigma_0 = 1.0$, $\sigma_1 = 2.0$, $H = 16$, $\alpha = 0.4$, $\eta = 0.8$, 5-sample window. (a) 4-layer network, 6-element training ensemble; (b) 4-layer network, 10-element training ensemble; and (c) 4-layer network, 40-element training ensemble.



97689-19

Figure 16. Continued.

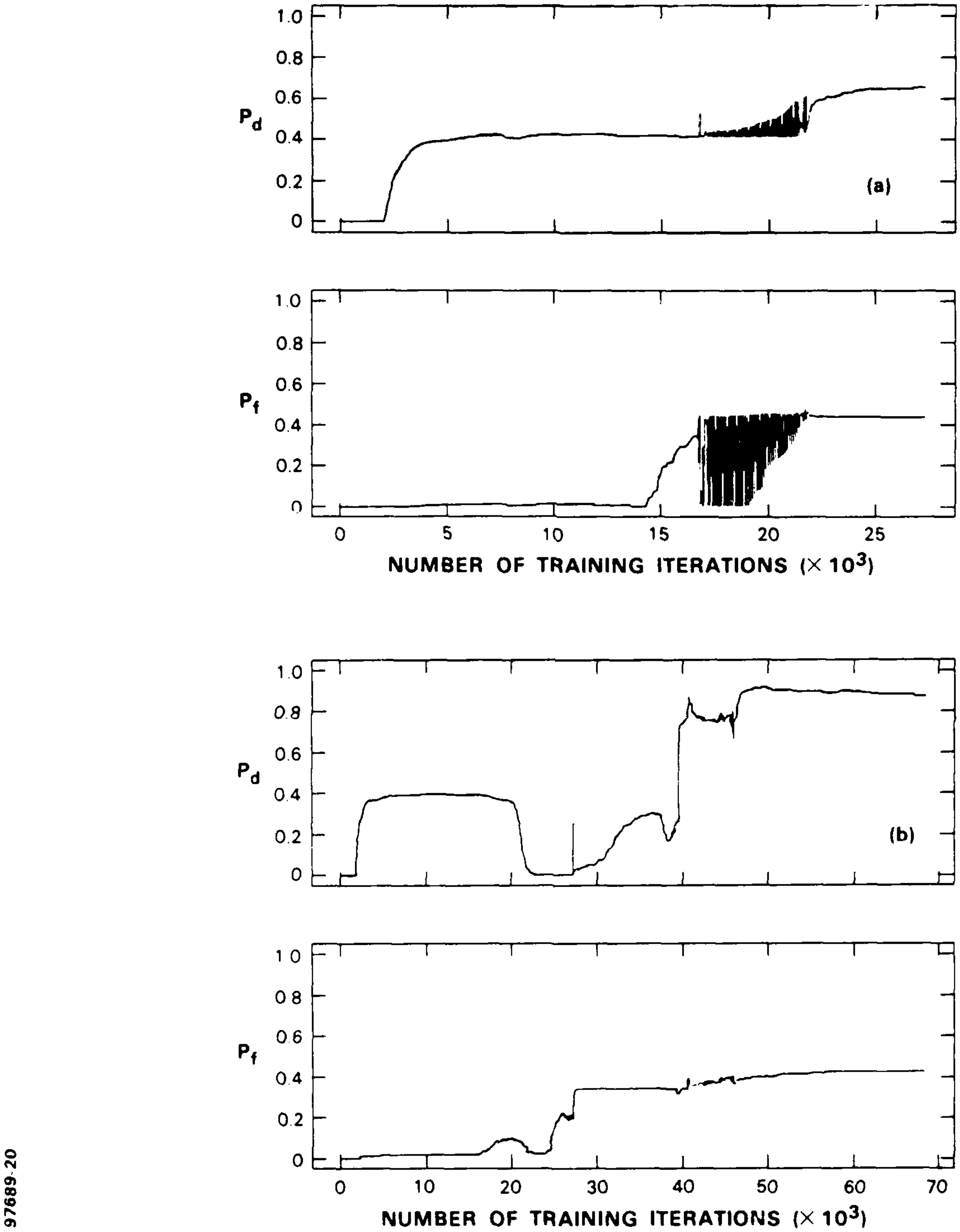


Figure 17. False alarm and detection probability vs iteration number; $\sigma_0 = 1.0$, $\sigma_1 = 2.0$, $H = 16$, $\alpha = 0.3$, $\eta = 0.3$, 10-sample window. (a) 4-layer network, 6-element training ensemble; (b) 4-layer network, 10-element training ensemble; and (c) 4-layer network, 40-element training ensemble.

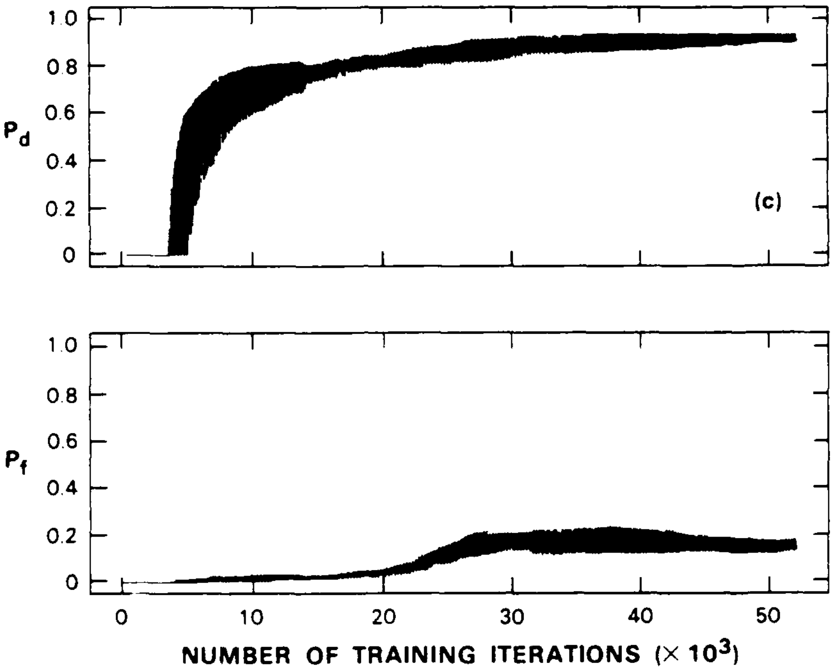
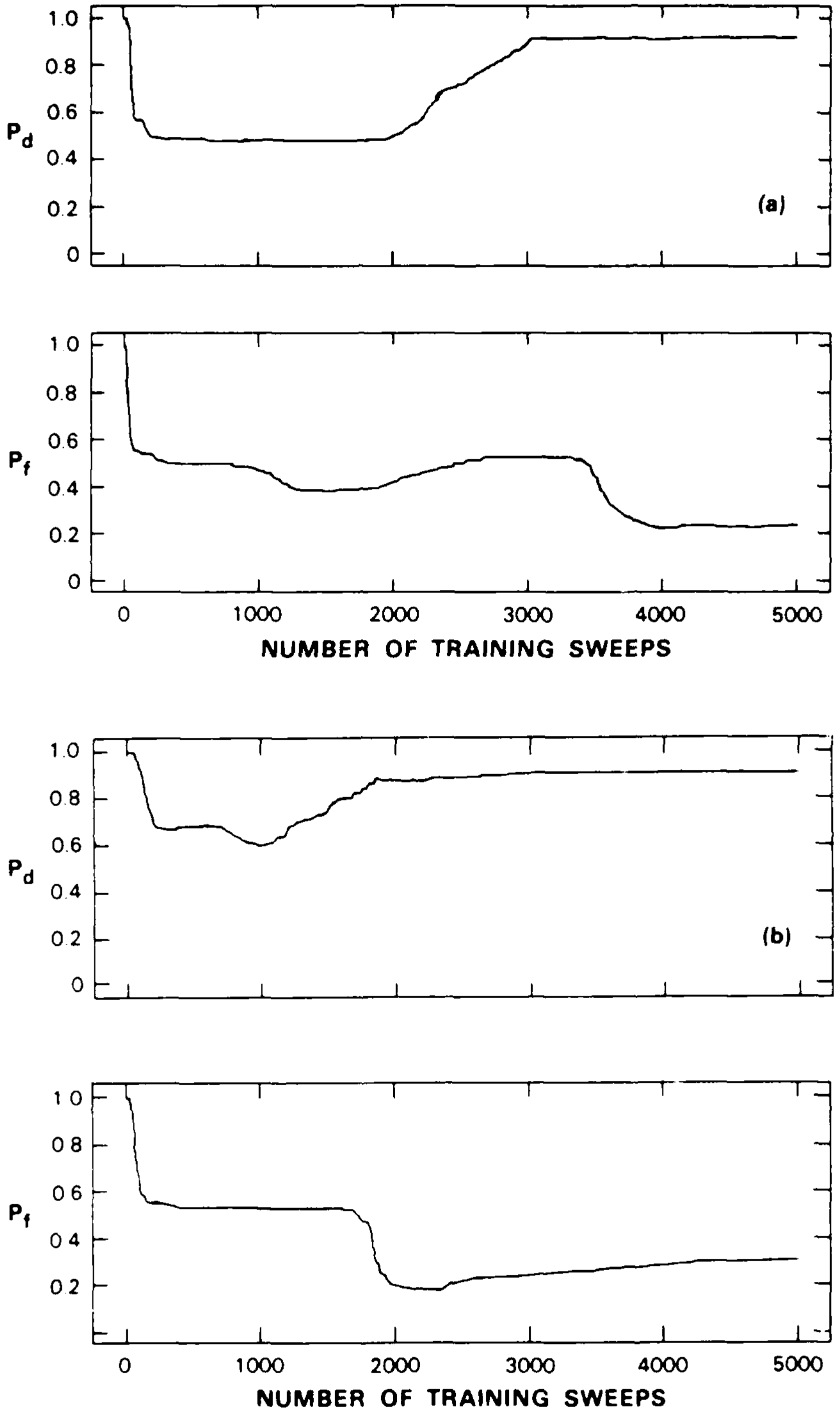
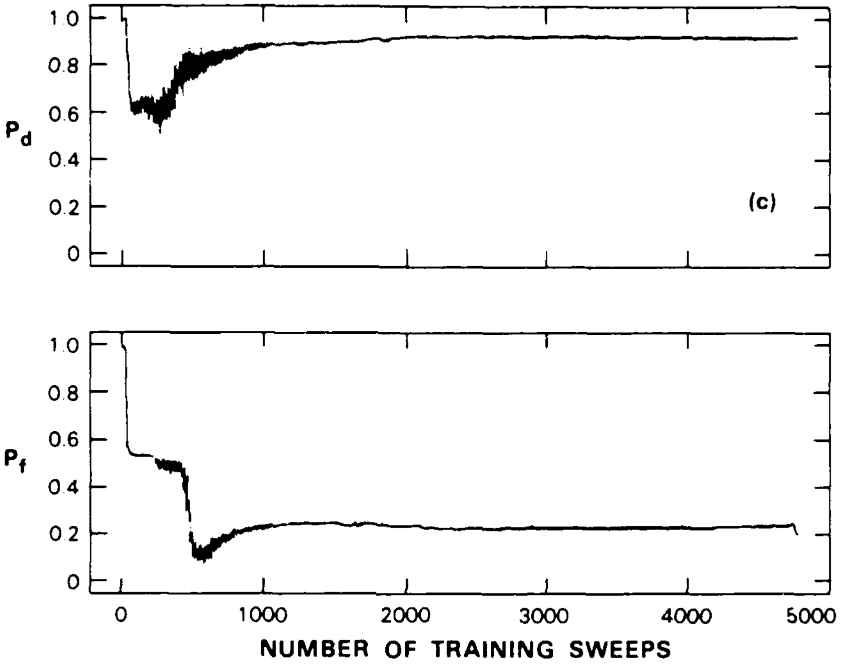


Figure 17. Continued.



97689-22

Figure 18. False alarm and detection probability vs iteration number: $\sigma_0 = 1.0$, $\sigma_1 = 2.0$, $H = 16$, $\alpha = 0.3$, $\eta = 0.3$, 20-sample window, scaled input. (a) 3-layer network, 6-element training ensemble; (b) 3-layer network, 10-element training ensemble; and (c) 3-layer network, 40-element training ensemble.



97689-23

Figure 18. Continued.

V. COMPARISON OF NEURAL NETS AND CLASSICAL PERFORMANCE

A specific goal of this report is a quantitative comparison of neural net and classical transition detection. A desirable feature of transition detection is the existence of a classical hypothesis test which forms the basis of the comparison.

For the case of the perceptron, the asymptotic values of P_f and P_d in Figures 10(a-d) to 12(a-d) are applied for the comparison with Neyman-Pearson testing. These probabilities are an average of P_f and P_d over a large number of iterations past the point in which the net has learned the map. Note that for small values of σ_1 and/or N, the mean probabilities have large variances. This is observed in the large iteration number limit of Figures 10(a) and (b); corresponding to σ_1 of two and windows of length two and five, respectively. The effect is due to input noise as reflected in uncertainty of the final connection vector A for the net. Note that variances in ensemble-derived probabilities vanish rapidly for higher values of σ_1 and/or larger window length.

The asymptotic false alarm and detection probabilities of each network were computed from ensemble-derived values in the limit of many training iterations. Figures 19 to 22 contain plots of the asymptotic perceptron and optimum classical values of P_f and P_d as a function of window length N. The Neyman-Pearson optimum thresholds were obtained from Figures 2(a-d) to 4(a-d). Standard deviations σ_1 of two and four were considered separately. The shaded region on each plot represents the range of false alarm and detection probabilities obtained from computed variances in these values. The thick curve represents the classical optimum as a function of the window length. Note that in all cases the averaged perceptron performance was close to the classical optimum, and the devia..on of the probabilities vanished at window lengths of twenty.

The performance of back propagation networks was considered with σ_1 of two for various window lengths. The simulations in Figures 16(a-c) to 18(a-c) contain net performance curves with 6-, 10-, and 40-element training ensembles for each (σ_1, N) pair. In Figures 17(a-c) and 19 performance is weakly dependent on ensemble size; suggesting that for sufficiently large windows ($N \geqslant 10$) distribution sampling is not a difficult problem. This is expected because the χ^2-distributions which form the input variances are not sufficiently overlapped to require dense sampling.

In Figures 16(a-c) it is observed that network weights and thresholds converged for all 5-sample window ensembles. The simulation in Figure 10(b) reveals that the perceptron failed to converge for 5-sample input variances. The larger number of adjustable parameters in the back propagation network may result in greater discrimination of overlapped distributions. Note, however, that while the asymptotic P_d of 0.74 is near the classical value of 0.78 [Figure 2(b)], a P_f of 0.54 is more than twice the classical optimum of 0.22 [Figure 2(b)].

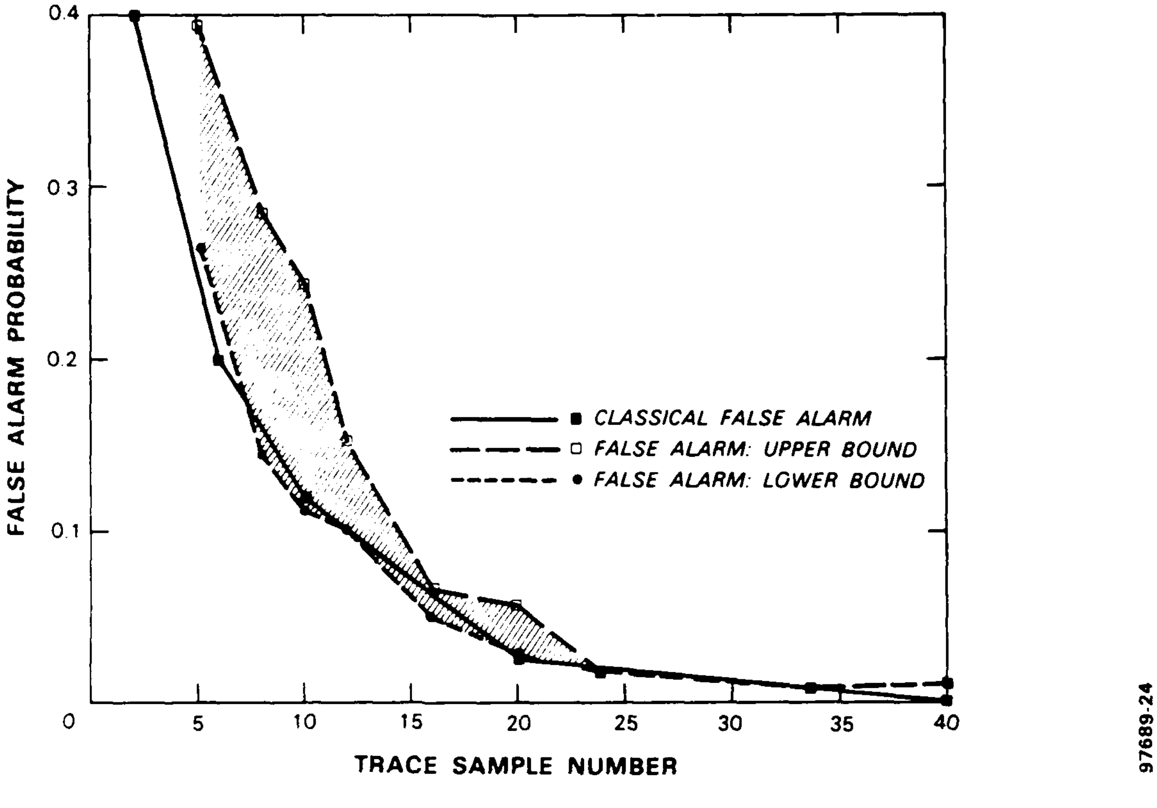


Figure 19 Perceptron and classical false alarm probability vs window length (N); $\sigma_0 = 1.0$, $\sigma_1 = 2.0$; shaded region represents range of perceptron performance from ensemble-derived probabilities; thickened curve represents classical performance.

The performance of the networks on larger window input variances is significantly closer to the classical optimum. Figures 17(c) and 18(c) have asymptotic (P_f, P_d) pairs of (0.16, 0.93) and (0.2, 0.93) for 10- and 20-sample inputs, respectively. These are to be compared with probabilities of (0.12, 0.88) and (0.04, 0.96) for the optimum Neyman-Pearson test with 10- and 20-sample inputs [Figures 10(c) and (d)].

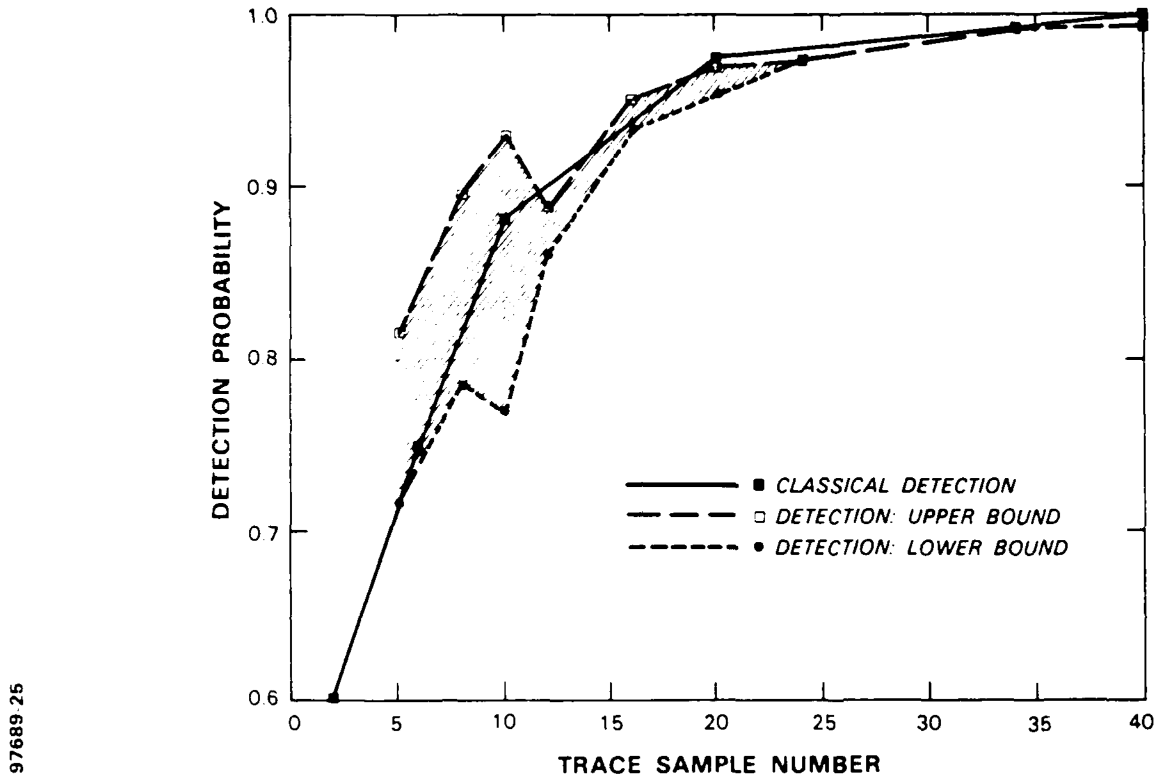
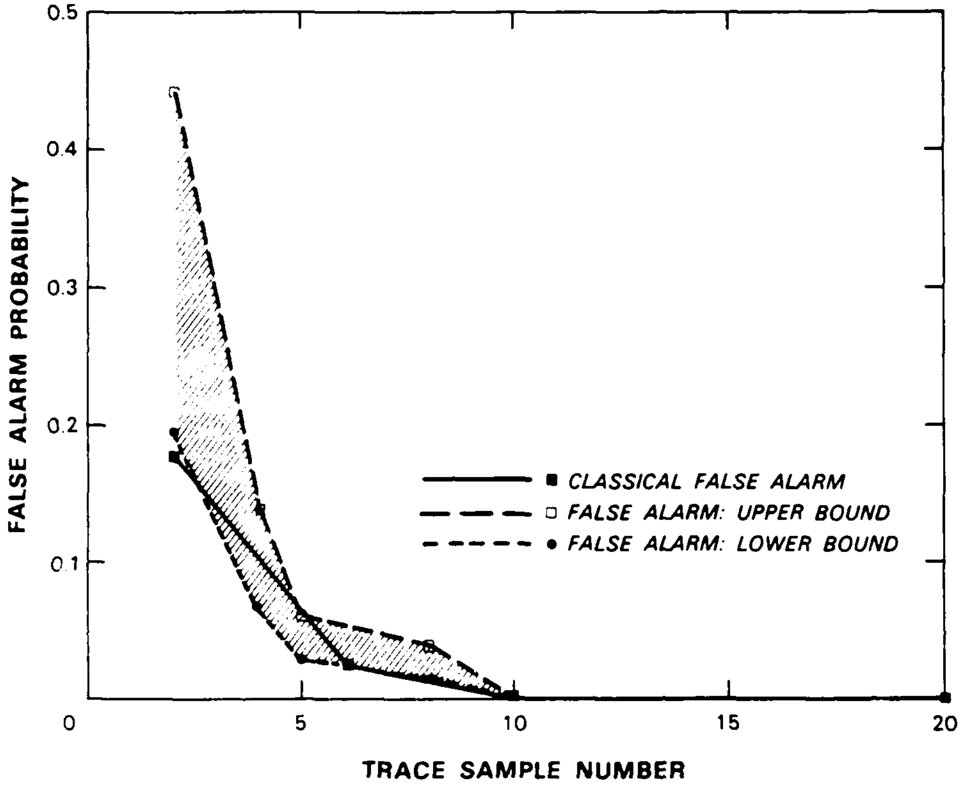


Figure 20. Perceptron and classical detection probability vs window length (N); $\sigma_0 = 1.0$, $\sigma_1 = 2.0$; shaded region represents range of perceptron performance from ensemble-derived probabilities; thickened curve represents classical performance.



97689-26

Figure 21. Perceptron and classical false alarm probability vs window length (N); $\sigma_0 = 1.0$, $\sigma_I = 4.0$; shaded region represents range of perceptron performance from ensemble-derived probabilities; thickened curve represents classical performance.

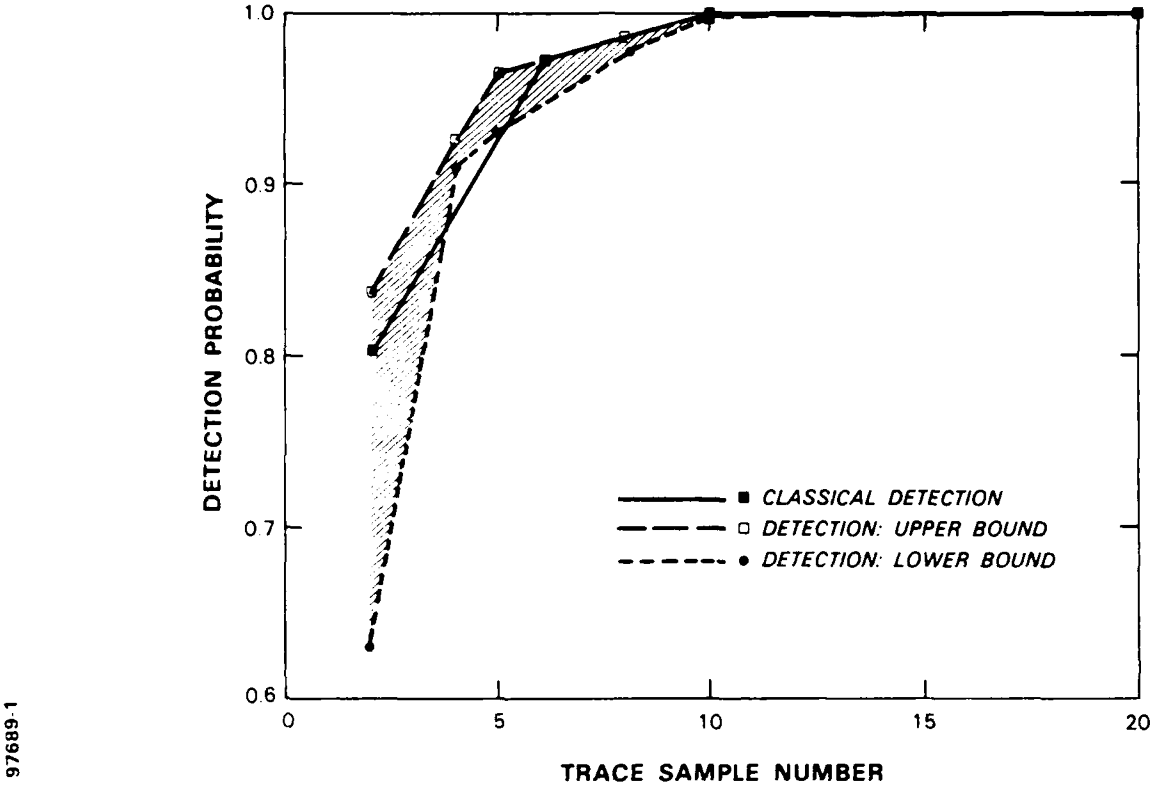


Figure 22. Perceptron and classical detection probability vs window length (N); $\sigma_0 = 1.0$, $\sigma_1 = 4.0$; shaded region represents range of perceptron performance from ensemble-derived probabilities; thickened curve represents classical performance.

VI. CONCLUSION

This report analyzed the performance of perceptron and back propagation neural nets on the stochastic exclusive-or problem. The test consists of the detection of a variance transition in Gaussian data from input of sample variances before and after the postulated transition. It is a prototype test; applications of which include maneuver detection from sensor data. For maneuver detection any data-derived sufficient statistic sensitive to target motion may be input to the network. For example, satellite maneuver detection has been performed by neural nets in which the input parameters are correlation lengths in the radar cross section. The only requirements for a statistic are parameter discrimination in the hypothesis domain and repeatability in the data.

As revealed in the previous sections, hypothesis testing Neyman-Pearson and neural net algorithms are radically different. Neural net testing requires a representative training ensemble. Although classical techniques are defined *a priori*, they are model dependent and optimum only at a particular threshold. Often the correct threshold is identifiable only for a stationary process, or if the nonstationarity is relatively simple (such as a variance transition). For this reason neural net decisioning may be more robust to noise properties. It should be noted, however, that neural net training requires the training ensemble to be representative of the continuous probability densities which describe the stochastic process. The problems of distribution sampling are likely to plague any ensemble-dependent learning algorithm.

In summary, it appears that for sufficiently discriminated input distributions a trained network performs near the Neyman-Pearson optimum. For example, Figures 19 to 22 show perceptron convergence to the optimum in <20 iterations. Important considerations for back propagation networks are the requirements of a large number of training iterations ($\sim 10^4$ for stochastic exclusive-or), and the tuning of training algorithm parameters. The required number of training iterations results from the large number of independent, adjustable parameters (connection weights, thresholds) in the network. This feature may aid net discrimination, as observed in network convergence for five sample input variances. The back propagation learning algorithm requires three added parameters; η for convergence rate, α for smoothing, and τ to control matching to binary output. A certain amount of experimentation was required to tune the parameters for optimum net performance.

ACKNOWLEDGMENTS

The author would like to thank Dr. Mitchell Eggers for stimulating discussions on neural network theory, and Mr. Timothy Khuon for skillful programming of the back propagation algorithm.

REFERENCES

1. D.E. Rumelhart *et al.*, Eds., *Parallel Distributed Processing* (MIT Press, Cambridge, MA, 1986). G.E. Hinton and J.A. Anderson, Eds., *Parallel Models of Associative Memory* (Erlbaum, Hillsdale, NJ, 1981).
2. D.E. Rumelhart, G.E. Hinton, and R.L. Williams, "Learning Internal Representations by Error Propagation," in *Parallel Distributed Processing*, D.E. Rumelhart *et al.*, Eds. (MIT Press, Cambridge, MA, 1986), pp. 330-334.
3. D.J. Willshaw, "Holography, Associative Memory, and Inductive Generalization," in *Parallel Models of Associative Memory*, G.E. Hinton and J.A. Anderson, Eds. (Erlbaum, Hillsdale, NJ, 1981); Y.S. Abu-Mostafa and J. St. Jacques, "Information Capacity of the Hopfield Model," IEEE Trans. Inf. Theory **IT-31**, 461-464 (1985).
4. R.J. McEliece *et al.*, "The Capacity of the Hopfield Associative Memory," IEEE Trans. Inf. Theory **IT-33**, 461-482 (1987).
5. M. Eggers, "Neural Network Application to Real World Decision Making," MIT Lincoln Laboratory, June 1986 (private communication).
6. F. Rosenblatt, *Principles of Neurodynamics* (Spartan, New York, 1962).
7. M. Minsky and S. Papert, *Perceptrons* (MIT Press, Cambridge, MA, 1969).
8. H. Van Trees, *Detection, Estimation, and Modulation Theory,* Part I (J. Wiley and Sons, New York, 1971).

APPENDIX
CLASSICAL TRANSITION DETECTION

In this appendix Equations (4) and (5) relating detection and false alarm probabilities to the quantities $\{p(i|j) \mid i,j \in (0,1)\}$ are derived. Recall that the index zero or one corresponds to noise deviation σ_0 or σ_1, respectively. The pair (i,j) denotes a transition from deviation σ_i to deviation σ_j, and the expression p(x| y) denotes the probability of x aetection conditioned on y. The relevant probabilities are then given by

$$P_d = p(\text{transition}|\text{transition}) \tag{A-1}$$

and

$$P_f = p(\text{transition}|\text{no transition}) \quad , \tag{A-2}$$

for detection and false alarm. The expression in Equation (A-1) is given by

$$p(\text{transition}|\text{transition}) = p[(1,0)|\text{transition}] + p[(0,1)|\text{transition}] \quad . \tag{A-3}$$

The relationship in Bayes Theorem,

$$p(x|y) = \frac{p(x,y)}{p(y)} \quad , \tag{A-4}$$

where p(x,y) represents the joint probability of x and y, can be applied to Equation (A-3) with the result

$$p(\text{transition}|\text{transition}) = \frac{p[(0,1),\ \text{transition}] + p[(1,0),\ \text{transition}]}{p(\text{transition})} \tag{A-5}$$

where p(transition) represents the prior probability of a transition. A transition is obtained by either a (1,0) or a (0,1) noise deviation pair. Equation (A-5) can be written in terms of the probabilities for specific deviation pair detection with the result

$$p(\text{transition}|\text{transition}) = \frac{p[(0,1),\ (1,0)] + p[(0,1),\ (0,1)] + p[(1,0),\ (1,0)] + p[(1,0),\ (0,1)]}{p[(1,0)] + p[(0,1)]} \tag{A-6}$$

where p[(i,j)] represents the prior probability of a deviation pair (i,j). Application of Bayes Theorem, Equation (A-4), results in the expression

$$p(\text{transition}|\text{transition}) = \frac{\big(p[(0,1)|(1,0)] + p[(1,0)|(1,0)]\big)\ p[(1,0)]}{p[(1,0)] + p[(0,1)]} + \frac{\big(p[(0,1)|(0,1)] + p[(1,0)|(0,1)]\big)\ p[(0,1)]}{p[(1,0)] + p[(0,1)]} \quad . \tag{A-7}$$

Recall that p[(i,k)|(k,l)] represents the detection of deviation pair (i,j) conditioned on the pair (k,l). Assuming that the decision for this occurrence is based upon a pair of maximum likelihood hypothesis tests before and after the transition, the conditional probabilities factorize; that is,

$$p[(i,j)|(k,l)] = p(i|k)\ p(j|l) \quad . \tag{A-8}$$

Substitution of Equation (A-8) into the conditional probabilities in Equation (A-7) results in the expression

$$p(\text{transition}|\text{transition}) = [p(1|1)\ p(0|0) + p(0|1)\ p(1|0)] \quad , \tag{A-9}$$

where p(i|j) is given in Equations (7) and (8). It is interesting that prior probabilities p[(i,j)] have canceled from Equation (A-7); indicating a detection probability independent of the prior distribution of deviation pairs.

The same argument applied to the false alarm probability in Equation (A-2) results in the expression

$$p(\text{transition}|\text{no transition}) = \frac{2\ \{p(1|1)\ p(0|1)\ p[(1,1)] + p(1|0)\ p(0|0)\ p[(0,0)]\}}{\{p[(0,0)] + p[(1,1)]\}} \quad . \tag{A-10}$$

In this case the probability depends upon the prior probabilities p[(0,0)] and p[(1,1)] for the ensemble upon which the hypothesis test is applied. An ensemble in which deviation pairs (0,0) and (1,1) are equally probable results in

$$p(\text{transition}|\text{no transition}) = [p(1|1)\ p(0|1) + p(0|0)\ p(1|0)] \quad . \tag{A-11}$$

Another interesting limit

$$p[(1,1)] << p[(0,0)] \quad , \tag{A-12}$$

corresponds to a high noise environment much less likely than a low noise environment. In this case Equation (A-10) reduces to

$$p(\text{transition}|\text{no transition}) = 2\ p(1|0)\ p(0|0) \quad , \tag{A-13}$$

independent of the probability p[(0,0)].

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
Technical Report 808

5. MONITORING ORGANIZATION REPORT NUMBER(S)
ESD-TR-88-125

6a. NAME OF PERFORMING ORGANIZATION
Lincoln Laboratory, MIT

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
Electronic Systems Division

6c. ADDRESS *(City, State, and Zip Code)*
P.O. Box 73
Lexington, MA 02173-0073

7b. ADDRESS *(City, State, and Zip Code)*
Hanscom AFB, MA 01731

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
U.S. Army Strategic Defense Command
Huntsville Sensors Directorate

8b. OFFICE SYMBOL *(If applicable)*
DASD-H-SBS

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F19628-85-C-0002

8c. ADDRESS *(City, State, and Zip Code)*
P.O. Box 1500
Huntsville, AL 35807-3801

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
31310F	80		

11. TITLE *(Include Security Classification)*
Neural Network Performance on the Stochastic Exclusive-or Problem

12. PERSONAL AUTHOR(S)
Robert Y. Levine

13a. TYPE OF REPORT
Technical Report

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Year, Month, Day)*
1988, July 8

15. PAGE COUNT
58

16. SUPPLEMENTARY NOTATION
None

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

neural network
hypothesis testing
transition detection
Neyman-Pearson

19 ABSTRACT *(Continue on reverse if necessary and identify by block number)*

The application of neural networks to the detection of variance transitions in Gaussian noise is analyzed. The problem is a benchmark example of hypothesis testing on a nonstationary stochastic process. Comparisons among perceptron, back propagation nets, and classical (Neyman-Pearson) decisioning is provided by Monte Carlo simulation. False alarm and detection probabilities for classical and neural net algorithms are computed and compared.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Lt. Col. Hugh L. Southall, USAF

22b. TELEPHONE *(Include Area Code)*
(617) 981-2330

22c. OFFICE SYMBOL
ESD/TML

83 APR edition may be used until exhausted.
All other editions are obsolete.